DOCUMENT RESUME

ED 297 703 IR 012 552

TITLE	Software Selection, Evaluation and Organization [and] Software Reviews. Article Reprints.
INSTITUTION	International Council for Computers in Education, Eugene, Oreg.
PUB DATE	[85]
NOTE	117p.; For a previous collection of reprints from the same source, see ED 262 757.
AVAILABLE FROM	International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403.
PUB TYPE	Collected Works - General (020) -- Guides - Non-Classroom Use (055) -- Journal Articles (080)
JOURNAL CIT	Computing Teacher; 1982-1986
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Software; Computer Software Reviews; *Computer Uses in Education; Elementary Secondary Education; *Evaluation Methods; *Instructional Material Evaluation; *Media Selection; Microcomputers
IDENTIFIERS	Boolean Logic; LOGO Programing Language; *Software Evaluation

ABSTRACT

This collection of reprints from The Computing Teacher contains 11 articles on the selection, evaluation, and organization of software published between August 1983 and March 1986, as well as more than 20 reviews of educational software packages published between December 1982 and June 1986. The articles are: (1) "The New Wave of Educational Software" (Ludwig Braun); (2) "How Do Teacher and Student Evaluations of CAI Software Compare?" (Barbara Signer); (3) "The DISC Model for Software Evaluation and Support Material Design" (Shelly Yorke Rose and Carol Klenow); (4) "Cooperative Learning: One Key to Computer Assisted Learning" (David W. Johnson and Roger T. Johnson); (5) "Teacher + Computer = More Learning" (Cathy Chmielowski Carney); (6) "Teaching with Your Fingertips!" (Sharon Burrowes); (7) "Selecting Computer Software--We Take It Seriously!" (Jean Donham); (8) "The Software Selection Process: Some Management Questions" (Anne Batey); (9) "Enrichment Courseware for Middle School Mathematics" (Lois Edwards); (10) "Doing Science" (Richard C. Adams); and (11) "Project DISC: Developing Indian Software Curriculum" (Carolyn Jacobi and Carl Edeburn). The computer software programs reviewed are designed for use in teaching problem-solving and reasoning skills; Boolean logic and basic elements of computer circuitry; reading skills; graphics programming; computer literacy; physics; science; social studies; writing; word processing; spelling; music; reading readiness; planning, writing and studying; animation, graphics, art; keyboarding; mathematics; robotics; Logo; and geometry. Other programs include grading systems, integrated applications, and readability analysis. Each review provides the name of the software program, the name of the reviewer, the producer's name, target audience, hardware requirements, and cost. (DJR)

ED297703

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ARTICLE REPRINTS

SOFTWARE SELECTION, EVALUATION AND ORGANIZATION

IR012552

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

David Moursund

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

The NEW WAVE of Educational Software

Guest Editorial by
Ludwig Braun

Introduction

Explorations of the computer as a tool to help students learn have been going on for about a quarter of a century. During most of that time most of the energy expended has been focused on pre-college students. Probably 90 percent of the available educational software addresses this group of students.

During this same period there has been a great deal of research on the computer's effect on rate of learning and retention of learned material. In both areas, the computer has had a dramatically positive effect, even though almost all of the research has focused on drill and practice applications. There also have been small efforts to assess the impact of learning to program a computer on the development of problem-solving ability, and the effective use of simulations—again with positive results. Despite these very positive research results, we have only begun to realize the computer's potential to help students. There are two reasons for this long gestation period:

- Until recently, computers were sufficiently costly that they were available to students only on a limited basis; and
- Only during the past three to five years have we learned enough about the development of creative learning environments to take advantage of the computer's great potential tial.

Until three years ago, most of the applications of computers in education consisted of teaching programming, or using drill and practice programs to teach facts. Recently there has been an exciting new wave of educational software, which I address here.

I wish to outline briefly some of the exciting ways in which this new wave permits the teacher to use the computer to improve the learning environments of students—ways which will make it easier for students to learn, more exciting for them to continue in the learning process, and more successful in pursuit of their goals.

The New Wave

The new wave of educational computing includes:

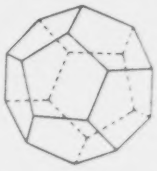
Learning to Write Well

Word processors, spelling checkers, outline generators, grammar and style analyzers and pre-writing guides will take much of the chore out of writing, permitting the student to concentrate on the *creative* part of writing, rather than the mechanics.

Discovery Learning

Since the early days of the Huntington Computer Project, it has been clear that the computer has a great deal to offer to educators as a flexible universe within which a "world" can be created with any set of characteristics desired. Simulations in the natural and social sciences have provided students with opportunities for experiential learning which are unavailable otherwise. There have been applications in biology, chemistry, engineering, ergonomics, physics, physiology and psychology to name just a few. In each of these fields, the teacher can create situations which conform to the normal laws of the universe (or to some subset—e.g., permitting the suspension of randomness), or if it is desirable pedagogically, which defy them. Students may be given control over system parameters so that they can explore their effects. Research has shown that this is very beneficial to students.

Relatively little work has been done exploring the effect of discovery learning in mathematics; however, this aspect offers much promise in enhancing students' understanding of mathematical concepts in algebra, calculus, statistics and probability, finite mathematics, etc.



For the first time, teachers of mathematics have available meaningful laboratory experiences for their students. Here, the student can discover and comprehend mathematical relationships and properties of mathematical functions in ways which have not been possible before. This appears to be especially valuable as a learning environment for students who have not succeeded in mathematics taught in conventional ways.

The relatively good graphics available on modern microcomputers have permitted dramatic improvements in the creation of discovery-learning environments in all disciplines.

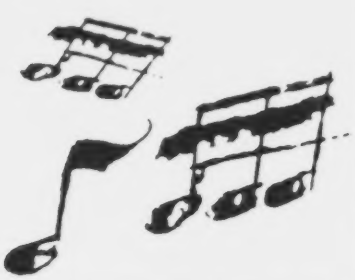
Copyright 1985 ICCE

Construction Sets

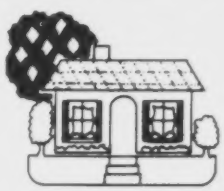
The construction set is a newcomer to the education scene. In such programs the user has a free-form environment with a set of objects which may be used to build simple universes. In the broadest sense, the construction set is a special-purpose language which permits the user to create complex effects without having to learn to program in BASIC, Pascal or machine language.

Rather than attempt to define construction sets in words, it seems best to define them by example. Some of the best of the currently available ones are:

- *Rocky's Boots* and its sequel *Robot Odyssey I* by The Learning Company provide the user with sets of pieces (including wires, AND, OR and NOT gates, actuators and sensors) from which rather complex logic circuits, and even large integrated circuits, may be constructed by unsophisticated users. These circuits are used in conjunction with sensors to control a set of actuators. *Rocky's Boots* was the first construction set to reach the market, although when it appeared the term was not in vogue.
- With the *Pinball Construction Set* by Bill Budge, the user is able to define a pinball machine by specifying how many bumpers, flippers, etc., there will be, and where they should be placed. Further, the user may specify the characteristics of the bumpers and the balls, so that varying amounts of energy are absorbed. It is possible to imagine a lot of good physics being displayed in this "world."



- *Music Construction Set* permits the user to "construct" a musical piece by arranging notes on a staff on the screen, and then specifying the characteristics of the instruments which will play the music.
- In *The Factory* by Sunburst the user can choose up to eight machines from an array which includes punch presses, rotating machines and paint striping machines. The user decides what the characteristics of the finished piece are and then selects the specific machines to use and their sequence to create the piece.
- *Just Imagine!* by Commodore permits the user to create stories which include background scenes, fixed objects (e.g., a gorilla), animated objects (e.g., a scuba diver) and the trajectory of each animated object, plus a 20-line story. When the user is finished, s/he can play the entire story back and enjoy an animated scene with background music and a story which scrolls onto the screen.

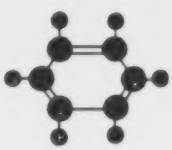


- *Dream House* by Joyce Hakansson Associates permits the user to take a bare house and choose colors to paint the walls, floors, ceilings and cabinets, then choose and arrange furniture in each room of the house. The user is also permitted to arrange trees, shrubs and flowers outside the house. Some of the objects (e.g., a flowing fountain or a canary) may be animated.
- *KoalaPad, Graphics Magician, Sorcerer's Apprentice,* etc., permit construction of graphic images from lines, rectangles, circles, etc. They also permit the creation of complex graphic images in full color, without the difficulties usually associated with drawing.
- *The Adventure Construction Set* by Electronic Arts lets you build your own adventure game using up to 500 characters, 250 settings ranging from the medieval era to science fiction, 30 sound effects, graphics and a variety of items including weapons, magical items and furnishings.

It is becoming clear that creative software developers can generate worlds in which exciting things will happen to students—assuming that construction set developers include creation of learning environments among their design goals. Most of the existing construction sets have as their motivation the generation of income (a worthwhile objective, if education is another), although some (most notably *Rocky's Boots*) were developed principally for education.

Some new construction sets I'd like to see are:

- A chemistry construction set which permits the user to build ionic and non-ionic compounds by arranging molecules in groups.



- A geometry construction set which contains lines, circles, ellipses, etc., to build geometric figures.
- A physics construction set to permit construction of experiments and physical worlds as the user desires them, with parametric values up to the user's discretion.
- Structures engineering sets to permit the building of structures with variable loading.
- A vector construction set which permits construction of vectors composed of several component vectors which may be added together algebraically. After the construction is complete, the computer should decompose the resultant and component vectors into horizontal and vertical components. The ability to rotate the entire set of vectors would also be desirable.
- Electric- and magnetic-field construction sets within which the user can create configurations of charges (or poles) and establish boundary conditions.

Adventure Games

Adventure games have tremendous potential for developing skills in reading, comprehension, decision making, problem solving and information processing. Unfortunately, most of the commercially available adventure games have been designed to be *commercial* successes, rather than good educational

© Copyright 1985 ICCE

tools. As a consequence, they have an unfortunate shoot-em-up, blow-em-up quality which turns off some students (most notably females), and which contributes little to the learning environment. It is possible to design such games so that they take advantage of the format and excellent parsers appearing on the market and still incorporate educational goals including appropriate vocabulary levels, sentence structure, story lines and graphics.

Laboratory Support

Modern microcomputers, combined with inexpensive microelectronic circuits (e.g., analog-digital converters), provide teachers with a new laboratory tool which permits substantial improvement in the students' experiences in gathering, digesting and displaying experimental data. It is now possible to combine an inexpensive computer with a low cost A/D converter to create a super-oscilloscope which gathers data, then processes them using statistical routines and displays the results graphically. This experience now is useful even with students who have little or no background in statistics (the technique has been used successfully with sixth graders).

"Probably the most important development in academic computing in the past five years is the emergence of tools—programs which convert the computer into a tool to carry out some job."

Telecommunications

Because of the development of low-cost modems, communications software, and national telecommunications networks, it now is possible to think in terms of providing students with access to powerful systems for communication with each other, with their teachers, with experts in any field, and with elaborate data bases. Such access permits students to participate in courses and other learning experiences, even if they are remote from the campus at which they are enrolled. Perhaps more importantly, the communication is asynchronous; it doesn't require users to be at the same place at the same time. This allows the student easy access to the teacher without regard to the schedules of the student or the faculty person, and without regard to the location of either. For the student with a difficult schedule (e.g., a student who travels a great deal), or for the student in a remote location, or for the student on campus whose schedule doesn't overlap that of the faculty person, telecommunications, properly applied, can be a good solution.

Tools

Probably the most important development in academic computing in the past five years is the emergence of tools—programs which convert the computer into a tool to carry out some job. We already have mentioned several tools above, including word processors, construction sets, telecommunications and laboratory-support systems. There are several others, among which the most important may be spreadsheet and data base programs.

Spreadsheet programs have become very popular over the past several years after the first, *VisiCalc*, was developed as a tool for financial analysts. They were developed around the conventional accountant's spreadsheet and permit the user to specify equations involved in developing financial statements. The operations specified by the equations are carried out automatically by the computer once the data has been entered. These programs have been touted as forecasters for financial people, permitting the asking of "what if" questions—i.e., they are general-purpose simulation languages. Inadvertently, their developers designed them so flexibly that they are useful as simulators in any discipline in which phenomena may be represented by sets of algebraic equations, even if those equations are nonlinear and time varying. Few have noticed, but a spreadsheet program can simulate phenomena in biology, chemistry, engineering, physics or physiology just as easily as those in the financial world for which they were intended.

Data base programs fall into two classes: those developed commercially, permitting the user to search for information and organize the selected subsets of information into useful forms; and those which are generic and permit the user to create personal data bases for a variety of purposes.

Epilogue

When we realize that none of the educational applications which I have described above existed in a meaningful way three short years ago, and when we reflect on the potential impact which these applications will have on education, we must stand in awe of the future. Who can project what the next three years may bring? Imagine what educational computing will be like in a decade hence—if you can!

[Ludwig Braun, New York Institute of Technology, Old Westbury, NY 11568.]

© Copyright 1985 ICCE

How Do Teacher and Student Evaluations of CAI Software Compare?

by
Barbara Signer

INTRODUCTION

Since the 1960s, the literature contains much research on computer assisted instruction. Traditionally, the studies have examined the following:

1. Effects of CAI on academic achievement.
2. Effects of CAI on attitudes toward subject matter.
3. Effects of CAI on attitudes toward computers.

More recently, studies have explored:

1. Needs for computer awareness..
2. Needs of the teacher in the CAI classroom.
3. Description and formative evaluations of computer projects in education.

Studies that compare student and teacher perceptions of CAI software are lacking. As an evaluator for a State of Florida grant for CAI research, I included this issue in the evaluation study.

DESCRIPTION OF CAI PROJECT EVALUATED

The grant developers proposed a unique alternative for the microcomputer as an instructional tool. It integrated the microcomputer into the conventional classroom in conjunction with the teacher's own teaching style—that is, the teacher controlled the pace and parameters of the program. One machine was required per classroom, and viewing was on a 19-inch screen. Interaction and opportunities for discussion were intended for total class presentation. The microcomputer was not used for every lesson, but only where it complemented rather than duplicated instruction by the teacher or texts. Therefore, instead of traditional usage of the term CAI, the project used computer integrated instruction (CII). The student data source consisted of students enrolled in all five Algebra II classes at a high school where CII was implemented. The teacher data source consisted of the two Algebra II teachers.

INSTRUMENTS

To obtain information necessary to evaluate the perceived effectiveness of CII, a variety of instruments were needed. If possible, an existing CAI evaluation instrument was used or modified; otherwise, one had to be developed specifically for this evaluation. The instruments used were the following: a) Signer Student Software Checklist, b) Signer Teacher Software Questionnaire and c) NCTM Teacher Software Evaluation Checklist.

Student Software Evaluation Checklist

The function of the *Signer Student Software Evaluation Checklist* (Figure 1) was to provide the evaluator with student feedback on the motivational and instructional value of the software used. This feedback included both teacher directed and independent student use. The criteria included for the instrument were: a) ease of administration, b) codability of items, c) opportunity for comments for each item and d) ability to identify strengths and weaknesses. While many questionnaires exist to solicit these kinds of information from students, all are intended for individual student use of computers only, and are not applicable for CII. In light of this, an original checklist was designed for this study. A content validity check of the instrument was accomplished by having specialists of mathematics education and educational evaluation review the items in the checklist for completeness of criteria and utility of information gathered through the instrument. The schedule of use of the checklist varied with the use of the software. Programs could be used one time or several times, by the class or by an individual. When instruction was presented to the class as a whole, two randomly chosen students from each class completed the checklist for each lesson. When programs were used by students individually, a checklist was completed by each student. Programs were evaluated at the time of use throughout the study.

Teacher Software Questionnaire

The aim of the *Signer Teacher Software Questionnaire* (Figure 2) was to collect teacher input regarding the value of the software for classroom presentation. The focus of the checklist included the following: a) classroom management, b) amount of teacher direction, c) amount of teacher preparation, d) suggested uses and e) a comparison to traditional instruction. The questionnaire included items that were open-ended so that software strengths and weaknesses could be identified.

Since this instrument evaluated a unique application of computer assisted instruction, the evaluator could not use available validated instruments to gather the information. A content validity check on

© Copyright 1985 ICCE

the newly developed instrument was conducted by having the two specialists, one in mathematics education and one in educational evaluation, review the overall questionnaire and determine whether items in the questionnaire solicited relevant feedback from teachers. Items were revised based upon reviewers' comments. It was used in conjunction with the Teacher Software Evaluation Checklist. Both forms were completed by each teacher after the initial use of the program.

NCTM Teacher Software Evaluation Checklist

The purpose of the NCTM Teacher Software Evaluation Checklist was to gather teacher feedback on the CII software. Criteria used to evaluate the programs were the following: a) appropriateness of level of instruction, b) diversity of modes, c) program flow and logic, d) motivational value and e) use of computer capabilities.

The *NCTM Software Evaluation Checklist* is contained in the National Council of Teachers of Mathematics (NCTM) publication, *Guidelines for Evaluating Computerized Instructional Materials*, prepared by Heck, Johnson and Kansky (1981). It contains items consistent with the criteria of the study and could be applied directly to the CII environment. The designing and field testing of the checklist were performed by a special NCTM Instructional Affairs Committee (IAC), chaired by Robert J. Kansky of the University of Wyoming. During this study, the checklist was completed by each teacher after initial classroom presentation of the software.

RESULTS

Teacher ratings were tabulated from the Software Checklist and the Teacher Questionnaire. Student opinion of each program reflected a consensus of the Student Questionnaires. The modal response for each question was recorded per program when clearly predominant. When the responses varied in both the positive and negative direction, the value assigned was recorded as mixed. Comments were noted when they occurred frequently on the evaluation instruments. Teacher ratings were slightly above student ratings for both instructional and motivational value. Teachers rated these criteria high across the programs. Students rated a small number either moderate or low on these criteria. There was a consistent difference between teachers and students when rating a program's pace. The teachers' ratings were favorable, while the students' were poor. Teacher-recommended changes were specific and unique to the particular program, while student-recommended changes were generalizable across programs, stressing more individual student use. Both teachers and students focused on motivational and visual qualities (graphics) for program strengths.

The reported differences between teacher and student evaluations can be attributed to their different roles. Teachers, as content specialists, are more critical of the specific content of a program. Knowledgeable in teaching strategies and the development of the concepts, they are aware of the varying pedagogical complexities for the different topics. Students, not cognizant of these intricacies, evaluate programs on the basis of interest, clarity and their level of participation.

LIMITATIONS

Conclusions of this evaluation study are not generalizable to all computer assisted instruction (CAI) studies. Computer integrated instruction (CII) is a unique application of computer assisted instruction. Conclusions are also specific to the environmental constraints of the school, which include fiscal resources and teacher personalities. These factors contribute to the effectiveness of teacher training. As teachers receive more and better training their perceptions of software will become more critical.

CONCLUSIONS

The natural implication of this study is that students and teachers have different perceptions of quality software, with the students being the stronger critics. Today's students are not naive about computer capabilities in terms of graphics and sophisticated peripherals. Their environment is filled with three-dimensional computer games with sound. Just as schools have had to compete with television, educational software must compete with the software found in arcades. As adults, we often assume that a microcomputer is motivating by itself. This is evident with the abundance of drill and practice and page-turning programs. Interactive programs, based on sound pedagogical development, require the feedback of both students and teachers. Extensive field testing in the schools is needed to move closer to higher quality software.

FUTURE RESEARCH

Educational researchers agree that formative evaluation is critical for successful development of innovations. This study attempted to answer the question: How do student and teacher evaluations of computer materials compare? The author suggests this and the following questions be included in future studies:

1. How do teacher and observer evaluations of computer use compare?
2. What are the teacher identified needs and levels of support to improve computer use?
3. What are the student perceived problems, their causes and recommendations?

The author will share details of this study. Those interested in future cooperative research that ad-

© Copyright 1985 ICCE

dress the above questions are encouraged to write the author.

REFERENCES

National Council of Teachers of Mathematics, *Guidelines for Evaluating Computerized Instructional Materials.* Reston, VA: National Council of Teachers of Mathematics, 1981.

Signer, B.R. *A Formative and Summative Evaluation Study of a Project Integrating the Microcomputer with Second Year High School Algebra Instructors,* doctoral dissertation, University of South Florida, 1981.

Figure 1.
Signer Student Software Questionnaire
[Write author for additional validation data and permission to reproduce forms.]

This will not be graded. Your comment will simply be used to improve your instruction. Your help in evaluating the instruction is invaluable.

Date _____ Evaluator's Name _____

Program Name _____

Chapter _____ Section _____

T_____ F_____ 1. I enjoyed the microcomputer program.
(Comments)

T_____ F_____ 2. Working with the screen became boring.
(Comments)

T_____ F_____ 3. I found myself just watching the screen, rather than trying to learn.
(Comments)

T_____ F_____ 4. I feel satisfied with what I learned.
(Comments)

5. The pace of the instruction using the microcomputer was:
_____ Just right _____ Too fast _____ Too slow
(Comments)

6. The amount of material covered during the lesson was:
_____ Just right _____ Too much _____ Not enough
(Comments)

7. I feel that the material presented was:
_____ Too difficult for me
_____ Just normal for me
_____ Perfect, I really learned quicker with the microcomputer
_____ Mostly a review since I already knew it
(Comments)

8. What would you change to make this lesson better for you?

9. What did you really like about this lesson?

Figure 2.
Signer Teacher Software Evaluation Checklist

Program _____ Microcomputer _____

Lesson Topic _____ Ch _____ Sec. _____

Teacher _____ Program Author _____

Times Presented _____

1. Was the pace appropriate for your instruction? (Why or why not?)

2. What would you like to see altered?

3. What suggestions do you have for teachers using it for the first time?

4. Was outside preparation required?

   a. How much time?

   b. Preparation of additional materials?

5. Compare this lesson using the microcomputer program to your past teaching of the topic.

   a. Student behavior

   b. Facilitation of material to be learned

6. General comments (hardware, classroom management, etc.).

7. Program use: Introductory? Drill practice? Review?

   a. Your use

   b. Suggested use

END■

Barbara Signer, University of Houston, Central Campus, College of Education, Department of Curriculum & Instruction, Houston, TX 77004.

© Copyright 1985 ICCE

The DISC Model for Software Evaluation and Support Material Design

by
Shelley Yorke Rose and Carol Klenow

Educators in Michigan now realize that they should preview and evaluate software prior to purchase.

Evaluation allows teachers to choose bug-free software which is based on accepted instructional principles and written for specific subject areas and grade levels. Previewing allows teachers to plan activities that complement the software and to devise ways to measure student learning.

While evaluation prior to purchase is a major step toward acceptance and implementation of computerized instruction in our schools, the process needs to be supplemented to enhance classroom use.

"The support material components were based on the DISC Project's philosophy that successful use of software required teacher planning, background information, pre- and post-computer activities and a record-keeping device."

The DISC (Documentation and Integration of Software into the Classroom) Project has increased the effective use of educational software in Oakland County by creating a model for instructional support materials to accompany evaluated educational software while training a network of 100 educators in its application.

This was accomplished by a two-phase approach. First, DISC participants were trained in the use of an adapted MicroSIFT evaluation form. They applied MicroSIFT's criteria to an educational software package purchased for them by the DISC Project. Educators then underwent the second phase of training in use of the DISC model. The software which participants recommended for classroom use was targeted for design of accompanying support materials which would facilitate teacher planning and increase its educational value to students.

The support material components were based on the DISC Project's philosophy that successful use of software required teacher planning, background information, pre- and post-computer activities and a record-keeping device. These requirements for effective use led to the design of the six components of the DISC support materials package.

Before using software, an educator should first assess whether available classroom time, setting and student abilities are compatible with a program's delivery of instruction. Next, the educator requires information regarding the program's educational content. In DISC's *Guidelines for Teachers*, this information is referenced in a one-page overview which precedes the program operation observations recorded in the *Program Operation Procedures* section.

Next, *Teacher Lesson Plans* encourage DISC participants to consider how the program would fit into a unit of instruction. Finally, the three remaining components are directed toward student use of the program. The *Preliminary Worksheet for Students* was designed to relate the computer activity to classroom objectives. This activity seeks to contravene the popular notion that computer activities do not reinforce traditional curriculum objectives. The *Reinforcement Worksheet for Students* is similarly designed to follow up computer use with activities such as class discussion or research projects. Finally, a *Computer Activity Worksheet* enables the teacher to monitor student interaction with the computer.

Participant feedback, the high quality of the support materials contributed to the *DISC Compendium*, and the continued interest in the DISC model have indicated the model's viability and acceptance both locally and nationally.

A similar project need not be expensive to implement. The majority of DISC's $15,000 funding went toward payment of substitute teachers while participants attended four 1/2 day sessions. The modest investment for software per participant (under $30) was generously repaid with support materials invaluable to many other software users.

The institution of the DISC model also saves money by preventing duplication of efforts. For us this meant that users of the more than 2,000 microcomputers in our 28 school districts could consult the *DISC Compendium* prior to software purchase and classroom implementation. These teachers then were provided not only with an evaluation, but also

© Copyright 1985 ICCE

with ready-made materials for adoption or adaptation *and* with a contact person experienced in the use of that software package. In an environment of experienced educators who have become wary of educational fads, support materials that relate software to curriculum objectives positively affect the establishment of the computer as an instructional tool.

"In an environment of experienced educators who have become wary of educational fads, support materials that relate software to curriculum objectives positively affect the establishment of the computer as an instructional tool."

A collection of evaluations and support materials (in this case, the *DISC Compendium*) simplifies individual curricular decisions while streamlining instructional use of microcomputers. This is especially significant in districts with many novice users to whom the computer's classroom role is not yet clearly defined. Just as new textbook users rely heavily at first on the instructor's manual, so do novice computer-using educators rely on software documentation. At times, experienced computer-using educators become novices when facing a new software package or new hardware.

A project adopting the DISC model should instruct educators in the importance of reviewing both evaluations by peers and reviewing commercial or teacher-developed support materials prior to judging the viability of a software package in their classrooms.

With DISC-based training to improve their software selection and implementation skills, educators will find that the microcomputer can be used at last to make a difference in the educational process.

[Ed. note: Carol Klenow and Shelley Yorke Rose are Instructional Computing Consultants for the Interactive and Instructional Computing Department (IICD) of the Data Processing Division of Oakland Schools, an intermediate school district serving 28 local school districts with over 200,000 students in Oakland County, Michigan.

For additional information on the DISC Project, contact the IICD-Oakland Schools, 2100 Pontiac Lake Road, Pontiac, MI 48054. Copies of the DISC Compendium *may be purchased through the IICD for $20. Also available is the* DISC Project Report and Training Manual *($10) which includes session outlines, hand-outs and transparency masters.]*

END■

© Copyright 1985 ICCE

Cooperative Learning: One Key to Computer Assisted Learning

by
David W. Johnson and Roger T. Johnson

There are three basic ways students can interact with each other as they learn. They can ***compete*** to see who is "best," they can work ***individually*** on their own toward a goal without paying attention to other students, or they can work ***cooperatively*** with a vested interest in each other's learning as well as their own. Students in cooperative learning situations [1]:

- **Celebrate** each other's successes
- **Encourage** each other to complete the assigned work
- **Discuss** the material being learned with each other
- **Help** each other
  - —Analyze and diagnose problems
  - —Transform information into other forms such as their own words, a picture or a diagram
  - —Relate material they are studying to previous learnings
- Are **motivated** by the enjoyable experience of working together
- **Learn** to work together regardless of individual differences

Cooperative Learning with Computers

In almost all schools the number of students far exceeds the number of computers, so it is inevitable that students will work with computers in small groups.

In a cooperative learning situation, students are assigned to groups of two to four and given an assignment to complete as a group, while ensuring that all group members master the material (no hitchhiking allowed) and interact skillfully.

When students work collaboratively at the computer, they can [1]:

- Observe and imitate each other's use of the computer, which increases their speed in mastering hardware and software.
- Observe, imitate and build upon each other's strategies, thereby increasing their mastery of higher-level reasoning processes.
- Experience the encouragement, support, warmth and approval of a number of classmates.
- Have peers evaluate, diagnose, correct and give feedback on their conceptual understanding, and orally summarize the material.
- Be exposed to a greater diversity of ideas and procedures, more critical thinking and more creative responses while completing the assignment.
- Have classmates encourage them to stay on task and exert concentrated effort.

"The combination of cooperative learning and computer assisted instruction has an especially positive impact on female students' attitudes toward computers."

Cooperative learning with computers (compared with competing with others or working independently at the computer) promotes more and better work, more successful problem solving, and higher performance on factual recognition, application and problem-solving

Cooperative Learning Groups

Positive Interdependence	Sink or swim together through: • Mutual goals • Division of labor • Role interdependence • Group rewards
Individual Accountability	It is verified that everyone learned.
Collaborative Skills Directly Taught and Used	Leadership, trust-building, conflict-management, decision-making, and other interpersonal and group skills have to be used.
Teacher Observes, Intervenes, Gives Feedback	Helps students to work together effectively and to understand the assignment.
Groups Process Their Effectiveness	Students gain skills in analyzing and improving how effectively they are working together.

© Copyright 1985 ICCE

STUDENT CHARTS

Cooperative Learning Rules

1. Work in groups of two to four.
2. Rotate roles.
3. Everyone must agree before information is entered into the computer.
4. Ask other group members for help when needed.
5. Summarize and explain to partner(s) what you are learning.
6. At the end of the lesson, briefly discuss how well you worked together and make recommendations for improving next time.

Roles/Responsibilities

Keyboarder	Enters group's decision into the computer.
Recorder	Records information from group decisions and from the computer. Everyone should take a turn at this job. Whenever possible all group members should record all the information.
Checker	Requires group members to demonstrate understanding and checks for group agreement.
Encourager	Encourages all students to participate by directly asking for their information and expertise and by reinforcing them for contributing.

© Copyright 1985 ICCE

tasks [2,3]. It is interesting to note that the combination of cooperative learning and computer-assisted instruction has an especially positive impact on female students' attitudes toward computers, while the combination of competition and computer-assisted instruction has an especially negative impact on female students' achievement, motivation, confidence in their ability to work with computers, attitudes toward computers, and attitudes toward the subject area being studied [2,3].

Cooperative learning groups may be used at the terminal itself or away from the computer to plan the material to be entered into the computer at a later time.

"Students in cooperative learning situations celebrate each other's successes."

Almost any learning task to be conducted at the computer can be structured as a cooperative learning activity [4,5,6]. Two examples may be found in the activity sheets which follow as well as a blank sheet for teachers to duplicate and fill in with their own cooperative learning activities.

Summary

The technology of computers and the interpersonal interaction promoted by cooperative learning provide complementary strengths. The microcomputer (depending on the software) does such things as presenting the task, controlling the flow of activity, doing computations, giving expert content and providing feedback. Within cooperative learning groups, students serve as models for each other, assist each other in analyzing and diagnosing problems, explain to each other the material being learned, teach relevant concepts and procedures to each other, keep each other on task, and share their satisfaction and sense of accomplishment with each other. Computer assisted instruction and cooperative learning is a partnership that maximizes the advantages of each.

[David W. Johnson and Roger T. Johnson, University of Minnesota, 202 Pattee Hall, Minneapolis, MN 55455.]

Footnotes

1. Johnson, D.W., & Johnson, R. (1983). The socialization and achievement crises: Are cooperative learning experiences the solution? In Bickman, L. (Ed.), *Applied Social Psychology Annual 4*. Beverly Hills, CA: Sage.
2. Johnson, R., Johnson, D.W., & Stanne, M. (1985a). Effects of cooperative, competitive, and individualistic goal structures on computer-assisted instruction. Minneapolis: University of Minnesota, manuscript submitted for publication.
3. Johnson, R., Johnson, D.W., & Stanne, M. (1985b). Computer-Assisted instruction: Influences of cooperative, competitive, and individualistic goal structures. Minneapolis: University of Minnesota, manuscript submitted for publication.
4. Johnson, D.W., & Johnson, R. (1975). *Learning Together and Alone: Cooperation, Competition, and Individualization*. Englewood Cliffs, N.J.: Prentice-Hall.
5. Johnson, D.W., Johnson, R., Holubec, E., & Roy, P. (1984). *Circles of Learning: Cooperation in the Classroom*. Alexandria, VA: Association for Supervision and Curriculum Development.
6. Johnson, D.W. & Johnson, R. (Eds.) (1984). *Structuring Cooperative Learning: Lesson Plans for Teachers*. New Brighton, MN: Interaction Book Company.

© Copyright 1985 ICCE

Cooperative Learning Lesson

SUBJECT AREA: Social Studies: Map-reading Skills (Grade 4)

I. DECISIONS

LESSON: Map Codes: Colors, Lines and Symbols

GROUP SIZE: 3

ASSIGNMENT TO GROUPS: Teacher-assigned: Heterogeneous (ability and sex)

MATERIALS: *Unlocking the Map Code*, Unit 2, Disks 1 & 2 (Rand McNally Electronic Programs, 1983); 3 logbook worksheets

II. SET THE LESSON

WHAT IS/ARE:

Academic Task:	Criteria for Success:
The group completes Activity #5 (Around the World)—one different trip for each group member.	Three logsheets will be combined for the group grade. Try to visit all continents and cross as many oceans as possible.

*Positive Interdependence:	*Individual Accountability:	*Expected Behaviors:
Group members are to reach consensus in planning and recording each of 3 trips. Summarize number of cities and continents visited and oceans crossed.	One group member is picked at random to explain how to plan a trip.	Trip Coordinator: Operates keyboard. Encourager: Offers support and acceptance. Checker: Checks for agreement on decisions.

III. *MONITORING

WILL BE DONE BY: ☒ Teacher ☐ Teacher/Student

FOCUS WILL BE ON: ☐ Whole Class ☐ Individual Groups ☒ Individuals

OBSERVATION SHEET INCLUDES THE BEHAVIORS OF:

Focus on individual skills: Taking turns at keyboard, offering encouragement, checking for agreement

*PROCESSING/FEEDBACK: Each student responds individually and then discusses with the group: What did you do to help your group this week?

Rate your group on how well they worked together: (Low) 1 2 3 4 5 (High) Explain reasons for your rating.

*An essential element of cooperative groups.

Cooperative Learning Lesson

SUBJECT AREA: Math: Locating points on a horizontal line, vertical line, or coordinate grid. Grades 2-6.

I. DECISIONS

LESSON: Hurkel, *Elementary Math Vol. 1* (MECC)

GROUP SIZE: 2

ASSIGNMENT TO GROUPS: Random

MATERIALS: Computer disk—*Elementary Math Vol. 1* (MECC)

II. SET THE LESSON

WHAT IS/ARE:

Academic Task:	Criteria for Success:
Locating points on a horizontal line, vertical line, positive coordinate system (0-10) and coordinate grid (−5 to +5)	Find Hurkel in 3 Tries: TERRIFIC! Find Hurkel in 4 Tries: EXCELLENT! Find Hurkel in 5 Tries: GOOD! Find Hurkel in 6 Tries: NEED PRACTICE!

*Positive Interdependence:	*Individual Accountability:	*Expected Behaviors:
Pair must agree on an answer before it is entered into the computer.	1. Each student must be able to explain strategy being used to find Hurkel. 2. Each student must take a turn operating program.	Rotate Roles: —Keyboard Operator —Explainer (Explains strategy being used) Both students make encouraging statements.

III. *MONITORING

WILL BE DONE BY: ☒ Teacher ☐ Teacher/Student

FOCUS WILL BE ON: ☐ Whole Class ☐ Individual Groups ☒ Individuals

OBSERVATION SHEET INCLUDES THE BEHAVIORS OF:

Taking turns, explaining, encouraging

*PROCESSING/FEEDBACK: Each student responds individually and then discusses with the group: What did you do to help your group this week?

Rate your group on how well they worked together: (Low) 1 2 3 4 5 (High) Explain reasons for your rating.

*An essential element of cooperative groups.

© Copyright 1985 ICCE

Cooperative Learning Lesson

Changing lesson plans to include cooperative interaction can be time-consuming at first. Here is a quick lesson plan worksheet which can be used initially to ensure all the critical elements of cooperative learning are incorporated into your lessons. As you use groups more often, this form can be used as a quick self-check.

SUBJECT AREA _____

I. DECISIONS

LESSON: _____

GROUP SIZE: _____

ASSIGNMENT TO GROUPS: _____

MATERIALS: _____

II. SET THE LESSON

WHAT IS/ARE:	Academic Task:	Criteria for Success:
*Positive Interdependence:	*Individual Accountability:	*Expected Behaviors:

III. *MONITORING

WILL BE DONE BY: ☐ Teacher ☐ Teacher/Student

FOCUS WILL BE ON: ☐ Whole Class ☐ Individual Groups ☐ Individuals

OBSERVATION SHEET INCLUDES THE BEHAVIORS OF: _____

_____

_____

*PROCESSING/FEEDBACK _____

_____

_____

*An essential element of cooperative groups.

© Copyright 1985 ICCE

The value of a good CAI program is more than just the skills it teaches; it can free up some of the teacher's time to concentrate on other things, providing students with more, richer learning experiences.

In the project described here, these teachers found that by turning over keyboarding instruction to a good CAI program, they could focus their own efforts on teaching higher-level thinking skills.

How well did it work? Very well: The students not only learned good keyboarding effectively and efficiently, but also developed better problem solving skills and learned to take responsibility for their own learning. A complete success, we'd say, and one with important implications that may reach into every classroom.

Educators often distinguish between two uses of the computer. On the one hand it is a tool for developing higher-level thinking skills which can foster such uniquely human abilities as goal setting, communicating with others and analyzing and evaluating situations. Software such as Logo, graphics and applications packages—word processors, data bases and telecommunications—implement the tool use of computers. On the other hand, the computer can be used for Computer Assisted Instruction (CAI). CAI, through tutorials and drill and practice, improves lower-level cognitive skills such as memory and comprehension. Our recent experience suggests, however, that higher-level thinking skills can also be developed through CAI.

The critical factor influencing what cognitive skills students develop using CAI is not the software, but the teacher. The teacher can structure the learning environment with CAI as the vehicle for students to develop goal setting, creative problem solving, analysis, evaluation and peer-interaction skills. CAI is compatible with this kind of environment and can be a powerful helpmate to a teacher trying to create this setting.

Higher-Level Thinking Skills

We learned how helpful CAI can be recently while teaching typing to our middle school students. Each seventh grade team of 100 students had been scheduled into the computer lab for several weeks. The first team used *Alphabetic Keyboarding* by Southwestern Publishing on the lab's 25 computers in rotation throughout the day.

The four team teachers and the lab coordinator had worked together to plan and carry out the activities, but, after this first group was finished, were only moderately satisfied with the results. So as part of the planning for the next team, the lab coordinator invited our school district's study skills coordinator, Bev Byington, to offer suggestions for improving the class.

Bev had just completed a sabbatical year studying techniques used by successful business people to formulate and achieve their goals. She helped us design a plan which incorporated these techniques for the second group of seventh graders. The outcomes from using her ideas changed our minds about the way CAI can be used with students.

The first and most obvious outcome was that the students learned more keyboarding skills in a shorter period of time than the previous group (figure 1). They also began to learn how to:

- Analyze and evaluate their own learning;
- Motivate themselves to achieve a desired goal;
- Work effectively with peers to increase their own rate of learning; and

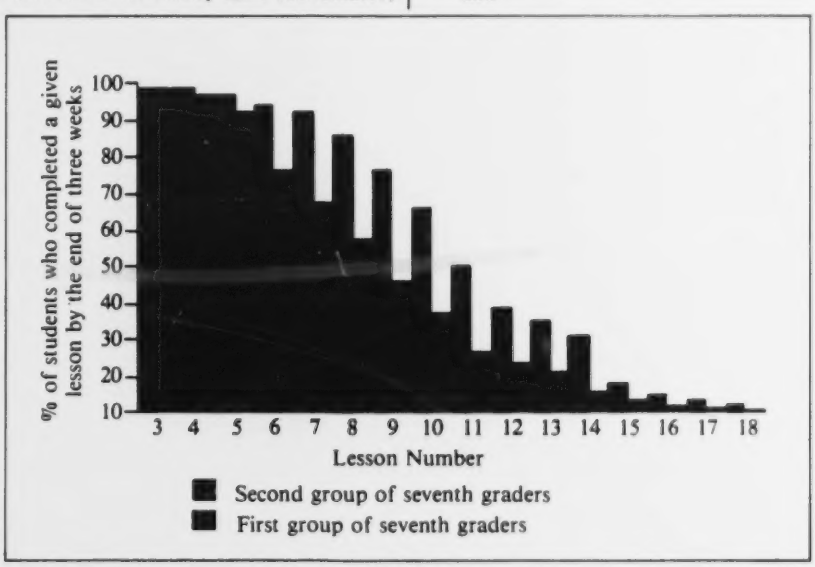


Figure 1.

© Copyright 1986 ICCE

Steps to Success

1. Work in SILENCE.
2. When you are frustrated, stop. Admit you are frustrated. Breathe deeply a couple of times. Look at the charts on the wall on which you have written why you want to learn keyboarding. Read through these "Steps to Success." Know that you will continue being successful on the computer after you take a break.
3. Read each screen on the computer carefully before pressing return to go on to the next screen.
4. Work through each section of the lesson in order, not skipping any, until you reach the point where the message appears, "Congratulations! You have finished this lesson."
5. After you have finished a lesson, and feel successful with it, go back to the section of that lesson which your success chart tells you to be tested on. Type two lines that contain the new material, with no more than one mistake. Then sign your name by that lesson on your success chart.
6. Have a friend or a teacher hold a paper over your fingers and watch you type two lines that contain the new material, with no more than one mistake. Then have that person sign their name by the lesson on your personal success chart.
7. When one particular letter frustrates you, return to the section that teaches that letter, and practice that section until you feel successful.
8. Work slowly, to be able to eventually work fast.
9. Anything worth doing is worth doing poorly. Making mistakes means you are trying to do something new and worthwhile.
10. Keep your eyes on the screen at all times.
11. Keep your fingers always on the correct home-row keys.
12. Feel the raised dots on the "D" and "K" keys when you need to find your place.
13. Know that you CAN do it!

Figure 2.

- Increase success through effective goal setting.

Why It Worked

How did CAI promote such skills?

Because good CAI contains clearly sequenced, individualized instruction, the teacher need not focus on the arduous task of providing carefully ordered, individually appropriate material and feedback for each student. Instead, s/he can concentrate on creating a learning environment which fosters the development of more globally important skills such as those listed above.

Because our keyboarding software could be used by each student to learn typing at his/her own pace, the teachers were able to concentrate on other goals. The goals we selected to work on were those Bev had suggested based on her work with successful business people.

Setting Their Own Goals

A primary goal was to help students learn to set their own goals. Before they ever entered the computer lab, we spent a period in their own classroom drawing out of them reasons why they would want to use this touch-typing program. As they gave their reasons, we wrote them, exactly as stated, on butcher paper visible to the whole class. The tendency for some students to be facetious disappeared quickly once they realized that we took their suggestions seriously and wrote down everything they said. As with any similar brainstorming exercise, people were piggy-backing on one another's ideas and suggesting reasons that we never would have thought of.

Each class' butcher paper was then posted in the computer lab for as long as that team was in the lab. It was con-

© Copyright 1986 ICCE

stantly in front of students' eyes as a reminder of the goals that they and their peers had set. This time spent helping them to "buy into" the goal of using CAI to learn to keyboard resulted in noticeably more commitment to this task than the first group exhibited.

Identifying Steps to Success

Once the students had committed themselves to the goal—learning to keyboard—the next step was to make sure they had the tools they needed in order to reach that goal. They needed to know that there were steps which, if followed, would assure them of success.

We began this process by distributing and discussing a short list of "steps to success in keyboarding" (figure 2) at our preliminary goal-setting meeting. These steps were culled from our experience with the first group of students, summarizing their comments about what had helped them learn keyboarding efficiently.

We used this list in several ways. The last five minutes of each keyboarding session was spent asking volunteers to share with the rest of the class which one of these steps they had found most personally helpful during the session. These testimonials encouraged both that individual student and those listening to believe in the validity of consciously using the success-building items on the list.

Students began to come up with additional steps to success as they started to notice what was helping or hindering them in their efforts to reach their goals. When they shared an original idea, other students would add it to their own lists. We then created a simple poster with the student's idea and displayed it in the computer lab.

At first only the bravest few in the class would volunteer during these sharing sessions at the end of a class. As the days went by, however, and they saw the contributions of other students being respected, even the most unlikely students began analyzing their personal learning techniques and sharing them with others.

Developing Good Peer Relations

These sharing sessions dovetailed nicely with another of our top-priority goals: helping students develop good peer relations. Another way we encouraged peer cooperation was to have students test one another at the end of each lesson. One student would hold a piece of paper over the hands and keyboard of the other student while watching the screen to see if a passage were typed correctly. S/he would then sign his/her partner's personal success chart, certifying that the partner had passed the test. We had been reluctant to initiate this peer-testing program at first, afraid that it would create considerable disturbance. But in fact, because they were convinced of the need of others for silence in order to concentrate well, they became very adept at testing one another without distracting those around them. Besides cultivating a cooperative attitude among students, this peer-testing program also facilitated individual conversations between students on how to improve keyboarding skills and freed the teachers from doing much individual testing.

Taking Charge of Their Own Learning

Yet another of our primary goals was to encourage students to take charge of

Personal Success Chart For _____

Lesson Number	Section To Be Tested On	Letters Learned In This Lesson		Sign-Off By You	Sign-Off By Friend	Sign-Off By Teacher
1	6	A S D F J K L ;				
2	5	Review				
3	8	E H O				
4	8	T I Left Shift				
5	6	Review				
6	6	Review				
7	8	N G Right Shift				
8	4	Review				
9	6	U C . P				
10	7	Y , X				
11	3	Review	GWAM			
12	7	M Z				
13	7	B ? Q				
14	4	Review				
15	5	V				
16	3	Speed				
17	3	Speed				
18	4	Speed				

Figure 3.

© Copyright 1986 ICCE

their own learning through the use of a personal success chart (figure 3). Students monitored their own success daily by charting their progress. As they successfully completed each lesson, they signed their initials next to it. As mentioned, a partner would then test the student on the lesson and sign his or her name as well.

Having the lesson sequence posted helped them see how their goals could be broken up into a number of objectives, each of which was possible to achieve. Their personal charts showed what had been already accomplished and what still needed to be done, focusing their attention on their daily goals and motivating them by showing what they had already accomplished.

Teacher Benefits

While the students were improving their abilities to understand, analyze, evaluate and share their learning techniques with one another, their teachers were discovering ways of encouraging this kind of student activity. We had time to do this because the computer was providing the drill and practice on an individualized basis.

We spent our time in a number of ways. One of the most enjoyable things we did was to create certificates—and publicly present them at the end of each class—for those who had completed a milestone lesson. We also had the opportunity to walk around the room, observing students quietly and finding one or two positive things to say to each individual during the period. Sometimes we worked with a student having a particular problem, becoming co-learners as together we analyzed and solved their problems.

From a teacher's standpoint, the environment was most attractive. Because the students felt successful about what they were doing they remained on task and discipline problems were almost non-existent. Instead we worked with students, individually and collectively, on techniques for increasing the effectiveness of their learning, and helped them focus on and celebrate the achievement of their goals.

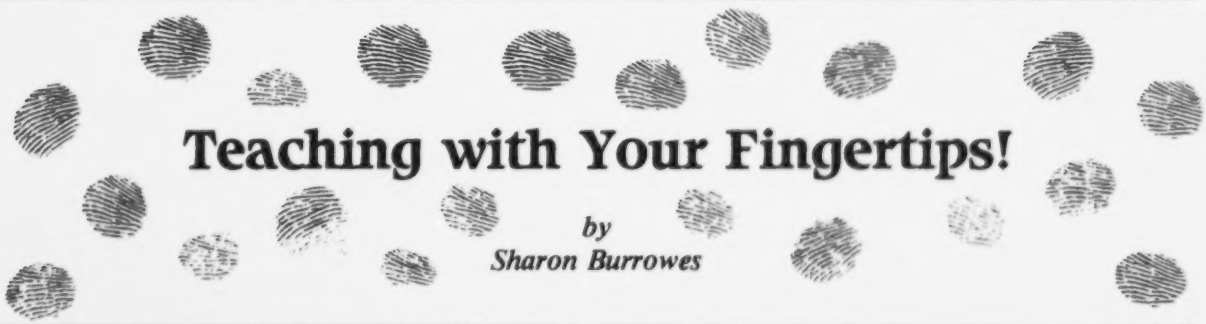
There were several surprises. One came when we asked students if, on Friday, they would like to take a break from the intensive keyboarding practice to see a speech-synthesis program demonstrated. Almost unanimously, they rejected the offer. When queried, they indicated that they had certain keyboarding goals they wanted to achieve and didn't want to have any time taken away from those efforts. We had not anticipated such a universally high level of commitment.

A few students had already taken typing classes and were able to move quickly and successfully through the lessons, reinforcing their skills. They then spent their time learning to use a graphics program to make special certificates for themselves (which we signed), and spent any remaining time exploring the program's extensive graphics capabilities. Thus the Promethean computer changed from a keyboard tutor to a graphics creator, providing a valuable way to individualize for students at the higher end of the spectrum.

This experience using a highly structured CAI program for keyboarding instruction can serve as a model of what will be increasingly possible in schools as good CAI for a variety of subjects becomes available for inexpensive hardware. With other subjects, of course, the steps to success would be different, the ways that peers could help each other would change, and the specific role of the teacher would vary. But the use of CAI as a way of encouraging students to develop higher-level thinking skills, to take responsibility for their own learning and to work cooperatively with their peers could be the same. The difference is in the teacher. Instead of taking power from the teacher, the computer can provide the teacher with an unprecedented opportunity to work with students to create an environment that supports an exciting and rewarding learning experience for everyone.

[Cathy Chmielowski Carney, Ryan Middle School, P.O. Box 1250, Fairbanks, AK 99707.]

© Copyright 1986 ICCE



Teaching with Your Fingertips!

by
Sharon Burrowes

A couple of years ago I had a chance to teach my first computing class using a large (25-inch) display screen hooked to my computer system. Before that I used the traditional blackboard and lecture approach. I was amazed at how much help seeing the computer "in action" was to my students as I introduced new material and answered questions. Since that time, I have found having a computer at my fingertips an invaluable aid in all aspects of teaching computing, and have encouraged others in our school system to use the computer in this way. Many of our elementary and junior high teachers bring in large monitors when teaching about the computer. In our high school we have a separate classroom with one demonstration system attached to a large color screen. This classroom, containing about 25 student desks as well, is used for all of our computer classes.

There are many ways you can profitably use the computer as a powerful teaching tool for groups of students. Some ideas and their advantages are described below.

Teaching Programming

- Students see immediately the results of any code you type. They don't have to "have faith" that the computer actually does what you say it does.
- Students see you make mistakes. This can be reassuring to perfectionists who think that every program should run the first time.
- You can respond to "What would happen if you. . .?" questions. As you change the code on the screen, you will find that students often answer their own questions.
- Students don't have to take as many notes on programs that you write. You can save them on disk and students can retrieve them later for study.

Teaching Applications Such as Word Processing and Data Bases

- The power of an applications program becomes more apparent. If you sort a list or print form letters in front of the class, students see how quickly and easily the computer performs such complex tasks.
- Relatively sterile lists of instructions for a particular task come alive. For example, students can see what happens on the screen as you make a global replacement in a document or search for a particular piece of information in a data base.
- By demonstrating common errors, you can help students avoid difficulties as they begin using a particular piece of software themselves.

Encouraging Discovery Learning

- Software that is designed to encourage problem solving, such as Logo or some commercial education games, can be used with a large group of students at once.
- You can teach techniques for brainstorming and apply them with an entire class.
- Students often spark ideas in each other, resulting in a more exciting learning environment.

Group Writing Projects

- Young children who have poor typing skills more easily become involved in creative writing when the teacher does the typing for all to see.
- Many students can become involved in such activities as writing a story or creating a poster.
- Students see and participate in the process of revising a first draft.

Giving Quizzes

- If you use a word processor to create quizzes, any unclear questions can be corrected immediately.
- Quizzes presented on the screen free the teacher from reading and rereading oral questions, and help the student who is not an auditory learner to perform better.
- When you go over an on-screen quiz there won't be any disagreement about the questions, because you can print a copy after you give the quiz.
- There won't be copies of the quiz if you don't make them, so the content is less likely to be passed from one class to the next. And you can easily make revisions for make-up work.

Once you decide to use a computer as a group teaching tool, you will find many uses that fit into your own classroom or style that aren't described here.

Some schools set up computer labs with networks that include a demonstration system hooked to individual student stations. I prefer using a single computer system with a large display screen because students seem to focus better on the material being presented; they are not distracted by the equipment and materials in front of them.

Copyright 1985 ICCE

If you are interested in using a large screen display, there are several pointers to keep in mind.

1. When you begin looking for display screens, check with your AV department and/or your local computer store to find out what connectors are needed and what display screens work best with the computer system you will be using.
2. Beware of extremely expensive display screens. Often the color monitor you use for showing videotapes will be adequate.
3. Although screens are improving, be sure your classroom is arranged so that all students are as close as possible to the screen. I have found that five rows is the deepest I can go with everyone still able to see a 25" color screen. Groups of five to 10 students can be seated in a semicircle around color monitors of the size standardly used with many computer systems. You will need to experiment with your computer system and display screen.
4. Eighty-column software is almost impossible to read on most large color monitors. If possible, use your software in a 40-column mode when demonstrating. An alternative is to hook several smaller 80-column monitors to your computer system and place them around the room.

Whatever arrangement you make, be sure to check the equipment with the computer system you will be using and a variety of software before you purchase. Outfitting a computer classroom can be expensive, and you should be sure everything works correctly before you buy.

[Sharon Burrowes, 807 College Ave., Wooster, OH 44691.]

Copyright 1985 ICCE

Selecting Computer Software—We Take It Seriously!

by
Jean Donham

The Iowa City Community School District serves 8300 students in 20 schools. The district developed a plan for instructional uses of microcomputers in 1982. The plan is being implemented in phases, based on available dollars and opportunities for teacher inservice.

One component of this initial plan allocated $5000 per year for the first three years to purchase software for evaluation by the selection committee and by the rest of the teaching staff. The district selected the *Evaluator's Guide for Microcomputer-based Instructional Packages*[1] as the evaluation instrument because it gives a comprehensive set of criteria.

The Process

The Superintendent directed the district library media coordinator to appoint a district-level committee to select software. This 20-member committee includes teachers, library media specialists and administrators. They were chosen because of their familiarity with computer-assisted learning and their knowledge of instructional design; the committee encompasses a range of content-area specialties. The district library media coordinator chairs the committee.

Teachers and library media specialists throughout the district suggest software for the committee to consider. If a title has already been previewed, a copy of the completed MicroSIFT evaluation form is sent to the person making the request. If a title has not been evaluated, the software is ordered on approval. (If a company refuses to sell on approval, their product is not considered for purchase.) Software is usually kept in the district on preview for one month.

When the software arrives, three committee members are assigned to evaluate the software. Evaluations are done individually in each evaluator's school so that students can be involved in trying out the software when appropriate. While effective software design is an important consideration, the evaluators also compare the objectives of the software to the district curriculum. This is an essential consideration and sometimes results in very well-designed software being rejected. (This has been a frequent problem, particularly in language arts and social studies.) Each evaluator uses the MicroSIFT criteria to assess the software and makes a recommendation either to purchase or not to purchase. The three evaluations are then reviewed by the coordinator, who in many instances also evaluates the software before making a final decision.

If approved, one copy of the software is purchased out of the central fund and is then available for preview by all schools through the District Media Center. All schools receive quarterly annotated lists of approved software. Schools may only purchase approved software (see Appendix).

If a software title is rejected by the committee, a copy of the completed evaluation form is sent back to the company along with the returned software. This practice provides producers with feedback. To date, over 275 instructional software titles have been previewed. Nearly two-thirds of the software evaluated has been returned.

Two *Core Software Library* lists, one for elementary and one for junior high (see figures 1 and 2) have also been developed by the software selection com-

Core Software Collection—Elementary Level
May 1984

Bank Street Writer
Scholastic

Basic Number Facts
Control Data

Delta Drawing
Spinnaker

Gertrude's Puzzles
Learning Company

Jigsaw
Island Software

King's Rule
Sunburst

Master Type
Lightning Software

Memory: A First Step
Sunburst

Teasers by Tobbs
Sunburst

Figure 1.

Core Software Collection—Junior High Level
May 1984

Applewriter for II + or IIe
Apple Computer, Inc.

Archaeology Search
McGraw-Hill

Bank Street Writer
Scholastic

The Factory
Sunburst

French Vocabulary Builder
Control Data

Lincoln's Decisions
Educational Activities

Master Type
Lightning Software

Nutrition and Nutrition Simulation
Health Education Services

Rocky's Boots
Learning Company

Spanish Vocabulary Builder
Control Data

Terrapin Logo
Terrapin, Inc.

Tobbs Learns Algebra
Sunburst

Typing Tutor
Microsoft

Word Attack
Davidson & Associates

Figure 2.

[1] *Evaluator's Guide for Microcomputer-based Instructional Packages*, MicroSIFT, Northwest Regional Educational Laboratory. Published by ICCE.

© Copyright 1985 ICCE

mittee. These titles were identified as appropriate for all elementary and junior high schools to own and use, based both on design and relationship to the curriculum. The list includes word processing and CAL software for content areas, and a Logo package. These libraries have been purchased for all schools in the district at approximately $600 per school.

Discussion

This approach to software selection has many strengths. First, it allows for independent evaluations. Evaluators work in their own schools rather than coming together to review software. This reduces the likelihood that one person's point of view will influence the decision making of another.

In addition, since evaluators work in their own schools to review software, they are able to try the software with students. Student interaction with a program often helps identify its strengths or weaknesses. This is especially true when considering ease of use, clarity of instructions, level of difficulty and readability of the program.

Software evaluation is a time-consuming task. Having a process coordinated at the district level has some benefits:

- The total amount of teacher time that might be spent in evaluating software is reduced. Teachers *not* on the selection committee spend their time reviewing software already identified as well-designed and appropriate. They can then move on to the task of determining how to use that software effectively with their students.
- A central file of evaluation results prevents repeated evaluations of the same software.

Saving teacher time carries with it some cost, however. Approximately 20% of the library media coordinator's time is spent on the software evaluation process. In addition, clerical time at the central office is required for ordering, returning and record keeping. Still, the process is efficient.

Responsible software evaluation is critical if teachers are expected to subscribe to the value of computer-assisted learning and if students are to benefit from it. Further, producers need to know that educators have high standards for acceptable software. Indeed, in Iowa City, we take software evaluation very seriously. **END**■

Appendix
Approved Instructional Software
Iowa City Community School District
May 1984

Software approved for use in this district has been reviewed by the selection committee on the basis of its sound design and its appropriateness for Iowa City's curriculum. Please note: The titles listed below do not include the titles in core libraries. The indicated grade levels are the grades at which the software is used in Iowa City.

Title	Producer	Subject	Level
Africa	Educational Activities	Social Studies	Grades 7 up
Algebra Drill & Practice I	Conduit	Mathematics	Senior high
Annam	Educational Activities	Social Studies	Grades 7-8
Atomic Structure	Opportunities for Learning	Science	Senior high
Basic Math Facts Drill	Random	Mathematics	Grades 4-8
Bumble Games	Learning Company	Mathematics	Grades 2 up
Bumble Plot	Learning Company	Mathematics	Grades 3 up
Catlab	Conduit	Biology	Senior high
Challenge Math	Sunburst	Mathematics	Grades 7 up
Chem Lab Simulation #2	High Technology	Chemistry	Senior high
Clock	Hartley	Telling Time	Grades 2-3
Color Keys	Sunburst	Problem Solving	Grades 4 up
Computer Math Games	Addison Wesley	Mathematics	Grades 1 up
Dasher	Conduit	Foreign Language	Senior high
Decimals	Control Data	Mathematics	Grades 7-8
Demographics	Conduit	Social Studies	Senior high
Demolition Division	DLM	Mathematics	Grades 4 up
Dragon Mix	DLM	Mathematics	Grades 4 up
Eat Smart	Pillsbury	Home Economics	Grades 7 up
Experiments in Human Physiology	Human Relations Media	Science	Senior high
Explorer Metros	Sunburst	Mathematics	Grades 7-8
Exploring Logo	Sunburst	Problem Solving	Grades 4 up
Factoring	MicroPower & Light	Mathematics	Grades 7-12
French Travel Vocabulary	Control Data	French	Grades 9 up
French Vocabulary for Shopping Use	Control Data	French	Grades 9 up
Fundamental Skills for General Chemistry	Opportunities for Learning	Chemistry	Senior high
Gas Laws & Kinetic Molecular Theory	Merlan Scientific	Science	Senior high
Geology Search	McGraw-Hill	Science	Grades 7-8
German Classroom Words	Control Data	German	Grades 9 up
German Vocabulary Building	Control Data	German	Senior high
Green Globs	Conduit	Mathematics	Grades 7 up
Incredible Laboratory	Sunburst	Problem Solving	Grades 3 up
Juggler's Rainbows	Learning Company	Spatial Concepts	Grades K-2
Kid Stuff I & II	Circadian	Computer Programming	Senior high
Le Francais Par Ordinateur —Paris en Metro	Heath	French	Senior high
Le Francais Par Ordinateur —Un Repas Francais	Heath	French	Senior high
Managing Lifestyles	Sunburst	Mathematics	Grades 7 up
Mastering the Parts of Speech	SVE	Language Arts	Grades 7 up
Math Ideas with Base 10 Blocks	Cuisenaire	Mathematics	Grades 2 up
Mathematics Activities Courseware	Houghton Mifflin	Mathematics	Grades 4-6
Meter Multiplication	DLM	Mathematics	Grades 4 up
Physics—Elementary Mechanics	Control Data	Physics	Senior high
Problem Solving in Algebra	Encyclopedia Brittanica	Mathematics	Senior high
Reading Comprehension I	Milton Bradley	Reading	Grades 5-8
Robot War	Muse	Computer Programming	Senior high
Sentence Combining	Milliken	Language Arts	Grades 4-8
Spanish Classroom Words	Control Data	Spanish	Grades 9 up
Spanish Travel Vocabulary	Control Data	Spanish	Grades 9 up
Spanish Vocabulary for Shopping Use	Control Data	Spanish	Grades 9 up
Speed Reader	Davidson & Assoc.	Language Arts	Grades 9 up
Step by Step 2	SVE	Computer Progamming	Senior high
Survival Math	Sunburst	Mathematics	Grades 7 up
Tutorial Quiz Master	Random	Interdisciplinary	Grades 3 up
Typing Tutor II	MicroSoft	Business	Grades 8 up
Waves and Vibrations	Merlan Scientific	Physics	Senior high
Word Master	DLM	Language Arts	Grades 3 up

[Jean Donham, Library Media Coordinator, Iowa City Community School District, Iowa City, Iowa.]

© Copyright 1985 ICCE

The Software Selection Process: Some Management Questions

by
ANNE BATEY

A survey of the literature on software selection reveals an abundance of carefully outlined criteria for thorough software evaluation. Although software evaluation is central, it does not complete the software selection process. Information on how to efficiently organize and manage the selection process is scarce. An exchange of such information would benefit any district that is developing a software selection process. This article poses three questions about management of the software selection process:

- Who is responsible for each step and decision?
- How is information transferred among the personnel involved?
- How long does the process take?

Following a discussion of the questions, an exercise is suggested to help you focus on the management aspects of your district's selection process. The purpose of the exercise is to generate a response from those of you in districts that have working answers to share.

Question 1: Who is (or should be) responsible for each step and decision in the software selection process?

A software selection process requires numerous steps and considerations. No single individual can be responsible for all of the decisions that have to be made before a package is purchased. The process is time consuming and should involve many district personnel who have a range of computer expertise and leadership roles. The personnel involved includes computer coordinators, computer facilitators, classroom teachers and students. The term "computer facilitator" refers to a variety of possible titles and roles—members of a software selection committee, in-building computer leaders, media specialists, teachers of educators, etc. In general, computer facilitators are the computers-in-education specialists. They often have considerable contact with classroom teachers, either because they continue to teach or because their jobs keep them within individual school buildings.

The question of respective roles in the process is refined by considering who is best suited for each responsibility. In general:

1. Computer Coordinators provide direction concerning decisions in the selection process that are administrative in nature, e.g., the budget, long-term planning, or district-wide computer curriculum.
2. The computer facilitators have technical expertise and can address the decisions and responsibilities that involve the implementation of educational software. Their contributions might include the following: educating teachers; researching software catalogs, reviews and evaluations; providing knowledge of the hardware capabilities; and screening for general software performance.
3. Teachers are in the best position to determine if a package meets their classroom needs and objectives. Final decisions that involve the instructional content and effectiveness should belong to the teachers who will be using the package.
4. Student response to a package also provides information for the decisions involving instructional effectiveness.

Question 2: How is information transferred among personnel involved in the software selection process?

Regardless of the size and the computer sophistication of a district, the selection process will generate lots of information. Since the process is not the sole responsibility of one person or group, considerable information flow and feedback must take place. How and by what routes does the information flow?

Question 3: How long does the software selection process take?

Many mistakes have been made in the haste to purchase software for newly-acquired hardware. A careful selection process reduces the waste of purchasing useless and ineffective software. But what is a reasonable time frame? How long do the individual stages in the process take?

Formulating Your Answers

The remainder of the article is an outline of the selection process. It summarizes some essential steps suggested by articles in computer education publications. The list is neither detailed nor comprehensive. It is included to encourage consideration of the management aspects of your own district's software selection process. In the interest of developing and sharing answers to the above questions, the following exercise is suggested:

© Copyright 1985 ICCE

Read through the steps and note the names or titles of the individuals that are responsible for each of the steps. Add, delete or change the sequence of steps as necessary. Some of the steps may involve a combination of personnel. Try drawing lines and arrows to clarify the flow of information through the process. Estimate the amount of time that each step requires. Perhaps the result of the exercise could be organized into a table or a flowchart. You are encouraged to submit your results. The information is needed and worth sharing.

I. **Preliminaries in the selection process:**
- Educate teachers: a software inservice, evening classes, etc.
- Collect resources: the software directories, catalogs, reviews.
- Develop schedule for the selection process.
- Determine budget limitations.
- Become well-versed in the district copyright policy or if necessary, establish one.

II. **Needs and objectives of your curriculum:**
- Develop a list of instructional needs to be met with computer use.
- Clarify needs, e.g., determine number of students involved per use, estimate desired length of involvement, specify the amount of teacher management necessary.
- Develop instructional objectives.

III. **Find software to match curriculum objectives:**
- Shop the catalogs with objectives, target audience, budget and hardware in mind.
- Read descriptions, reviews and evaluations.
- Decide on an acceptable package.
- Obtain preview copy: order from publishers who grant preview periods or provide demo disks.
- Ensure that preview rules and deadlines are followed.

IV. **Initial Software Preview:** A technical screening for performance in areas that do not involve instructional content. This includes:
- Complete user support materials
- Trouble-free operation, well-designed screen layout, sufficient user control
- Descriptive error messages with branching options
- Appropriate use of computer capabilities.

V. **Preview of software content:** A screening for instructional content. This includes:
- Topic and objective are clear in program and documentation.
- Program objective(s) match desired objective(s).
- Computer availability will not limit effective use of package.
- Management demands on teacher will not limit effective use of package.
- Students learn from the package.
- Prerequisites and interactions are appropriate to student ability and maturity.

VI. **Final decisions and purchase:**
- Compile results from previews.
- Select the winning package

Bibliography

A listing of the more comprehensive and useful readings.

1. Cohen, V.B., "Computer Courseware Development and Evaluation." *Educational Technology*, 23:1 (Jan. 1983), p. 9-14.
2. EPIE & Consumers Union, "No Preview, No Purchase." *The Computing Teacher*, 10:5 (Jan. 1983), pp. 35-36.
3. Forman, D., "Courseware Evaluation for Curriculum Integration." *AEDS Monitor*, 20:10, p. 10.
4. Jostad, K., "Search for Software." *AEDS Monitor*, 20:10, pp. 25-26.
5. Lathrop, A., Goodson, B., *Courseware in the Classroom*. Menlo Park: Addison-Wesley Publishing Co., 1983.
6. MicroSIFT, *Evaluator's Guide*. Eugene: ICCE, 1983.
7. Neumann, R. "A District Software Evaluation Plan that Works." *Electronic Learning*, Oct. 1982.
8. NCTM, *Guidelines for Evaluating Computerized Instructional Materials*. Reston, VA: NCTM, 1981.
9. Slesnick, T., "Hold It: You're Using Computers the Wrong Way." *The Executive Educator*. April 1983, pp. 29-30.
10. Steinberg, E., "Reviewing the Instructional Effectiveness of Computer Courseware." *Educational Technology*, 23:1 (Jan. 1983), pp. 17-19.
11. Olds, H. "Evaluating the Evaluation Schemes." *Evaluation of Educational Software: A Guide to Guides*. Chelmsford, MA: NEREX, Inc., 1983.
12. Watt, M. "Making a Case for Software Evaluation." *The Computing Teacher*, 9:9 (May 1982), pp. 20-22.

[Anne Batey is a graduate student in computer education at the University of Oregon. She has taught high school science in North Bend, Oregon for the past five years.]

© Copyright 1983 ICCE

Enrichment Courseware for Middle School Mathematics

by
Lois Edwards

Computer courseware is a natural resource in middle school mathematics classes. Since middle school mathematics students possess widely differing degrees of motivation, skill in basic arithmetic, experience in applications and problem solving, and maturity toward abstract reasoning, teachers should try to provide individualized instruction over a large range of topics. Fortunately, middle school mathematics classes, ranging from remedial arithmetic to first-year algebra, are often less tightly structured than the elementary arithmetic curriculum that precedes them or the traditional algebra and geometry sequence that follows. This gives teachers somewhat greater flexibility in selecting content, ordering topics of study, and varying the delivery of instruction. Courseware can effectively deliver traditional content and can effectively introduce students to math concepts not otherwise accessible.

Motivation is of primary concern at this grade level, both for students who have not yet mastered elementary arithmetic and for talented students who may be bored with arithmetic. Courseware can provide engaging learning experiences for both of these groups, as well as for average students.

Tutorial and drill courseware with traditional content has been well-publicized, so this article concentrates instead on representative programs that address problem solving, applications, number theory, probability, number systems, graphs and pre-algebra topics.

PROBLEM SOLVING

For best results, problem-solving courseware should be integrated into a problem-solving unit with other learning activities such as teacher presentations, manipulatives, class discussions, etc.

Problem-Solving Strategies, from the Minnesota Educational Computing Corporation (MECC), contains four programs to teach the strategies of trial and error (guess and check), exhaustive listing and simplifying the problem, along with tools such as making a chart and looking for patterns. The content consists of mind-teaser, puzzle problems: How many squares are there in a 5x5 grid? (not just 25!) How can one efficiently color a map? What relationship exists between the size of a pool table and the path of a ball on it?

Two tutorials teach the three strategies directly with color graphics and a high degree of student interaction. Then, in *"Thinking With Ink,"* students try to color a map at the lowest possible cost with paint colors that have different prices. Simplified warm-up maps and complex challenge maps are provided. Students will keep coming back to improve their previous scores by devising even better strategies (there is no "right" answer). The fourth program, *"Pooling Around,"* a simulation of a pool table with one ball, enables students experienced in problem solving to collect data and find a variety of patterns relating the pool table dimensions to the path of the ball. This is best used as a small group activity, so that the exchange of ideas and verbalization of hypotheses increase the instructional value.

The Factory, by Sunburst Communications, focuses on the strategy of working backwards. After experimenting with a variety of simulated machines which punch holes, paint stripes and rotate objects to discover their effects on a square of metal, students are challenged to design a series of machines that will duplicate a computer-selected finished product. In another option, a student can specify a finished product and challenge classmates to create the required machine. Excellent color graphics contribute to the intrinsically motivating, deductive-thinking exercise.

Another package from Sunburst Communications, *The Pond,* works with pattern recognition—patterns of lily pads in a pond with a pad-hopping frog who tries to stay dry. While the simplest levels of this program are easy enough for young children, the most advanced will challenge adults. This package provides practice in finding visual patterns (two pads up, three left, etc.) that should help students see numerical patterns in mathematical problems.

Rocky's Boots, from The Learning Company, combines problem solving and creative thinking with a content area not normally found in middle school—logic gates (and, or, not). Excellent color graphics and a high degree of student control create a motivating environment in which students devise machines using wires and logic gates, employing both analysis and synthesis while they enjoy the games. Four programs teach the skills needed to play the two games included in the package. In a mathematics class, this courseware could be tied into studies of Venn diagrams or formal logic. It is good preparation for computer programming.

Deductive thinking is the basis for two educational math games that have demonstrated their effectiveness over the years. In *Bagels* the program selects a random two- or three-digit number which the student tries to guess from clues about correct digits and placement. Interpreting the meaning of the clues and selecting the best guess is both intellectually challenging and fun. *Taxman* requires an understanding of factors of whole numbers and provides a challenge to students to test their knowledge of primes. The Taxman gets the factors of the numbers the student chooses, so students do a great deal of "what if" thinking to devise a strategy to beat the Taxman. *Bagels* and *Taxman* are found on MECC's *Elementary Volume I* and in numerous other collections.

In *The King's Rule,* from Sunburst, students form hypotheses to deduce what rule fits a given set of three numbers. The six levels of difficulty provide an appropriate challenge for any student. In creating several hypotheses for a given problem, students must distinguish between what must be, what could be and what cannot be. This highly addictive program keeps groups of students (and teachers) thinking and

© Copyright 1985 ICCE

debating their hypotheses until someone turns off the computer.

Magic squares and related arithmetic puzzles have been used for many years to stimulate pattern recognition. The computer program *"Magic Squares"* in the *Arith-Magic* package from Quality Educational Designs (QED), guides students through explorations of averages and sequences. Two additional problem-solving games involving subtraction and multiplication complete the package. Magic square puzzles and similar challenges can be generated and printed on paper using MECC's *Computer-Generated Mathematics Materials, Volume I.*

APPLICATIONS

Applying arithmetic and problem-solving skills to real-life problems is aided by several different types of courseware. One such aid is simulations, like the economic simulations *Sell Lemonade* and *Sell Bicycles* (MECC's *Elementary Volume 3).* Simulations present realistically complex situations using random numbers and the fast and accurate calculations of the computer. Students can practice interpreting mathematical data, analyzing situations and deciding on a strategy to solve the problem. The computer alleviates the burden and frustration involved in complicated arithmetic calculations.

The simple database in MECC's *Polls and Politics* courseware enables students to collect data on a questionnaire of their own design, and then enter it into the computer. The program calculates and displays frequencies of responses and cross tabulations in both raw data and percentage figures. Students learn to read, interpret and analyze data through a realistic example of the value of percentages in a statistical setting.

Word problems, a perennial source of difficulty for middle school students, can be investigated and solved using an innovative program called *SemCalc* (semantic calculator) from Sunburst Communications. Students enter each item given in the problem, specifying *how much* (numerical value) and *what* (unit of measure). Then they decide what operation to perform. The program displays the units of the answer produced by the student's choice of operation. If these units don't match the desired ones, the student knows to try a different operation.

NUMBER THEORY, PROBABILITY, NUMBER SYSTEMS, GRAPHS

Factors and primes, important topics in working with fractions and later in algebra, are given a mathematically sound and highly interactive treatment in the *Factoring Whole Numbers* courseware package from Quality Educational Designs (QED). The novel format of a tutorial and a game or exploration activity for each topic (12 programs in all) provides a motivating, concrete approach to factoring concepts, including the Sieve of Eratosthenes, twin primes and Goldbach's conjecture. The exploration programs provide challenges in discovering patterns in number theory.

Two other QED packages, *Fractions* and *Decimals,* use this same tutorial in a game/exploration format and are designed to follow in sequence. They encourage the use of manipulatives along with graphic computer tutorials. *Decimals* includes games involving decimal/binary conversion.

Computers are a valuable tool in teaching probability because of their ability to simulate random events and to perform large numbers of calculations quickly. MECC's *Probability* courseware includes both simple coin simulations and a surprising game activity that motivates a deeper study of expected events in rolling dice.

While number systems of bases other than ten have long been a topic of study because they increase understanding of our base ten system, computers provide a real-life example where other bases (two and sixteen) are important. The *Number Systems* package from MECC includes complete lesson plans plus computer programs to provide practice in converting between bases in a game setting.

MECC's *Graphing* courseware combines the math topic of graphs with the computer's graphics capabilities, creating a bridge between math and programming concepts.

PRE-ALGEBRA TOPICS

Discovering Equations, from Sunburst Communications, uses balance scales to introduce the concept of solving simple linear equations, demonstrating the effectiveness of the computer as a manipulative in teaching abstract concepts.

Another Sunburst product, *Tobbs Learns Algebra,* has large enough print to be used effectively as a whole class demonstration activity. Students find numbers to fill in a chart according to a given relation, or they produce the relation to match the given numbers.

The popular game *Hurkle,* found in MECC's *Elementary Volume 1,* has been widely used to strengthen the concept of ordered pairs as points on the coordinate grid.

• • •

Middle school mathematics is rich in motivating, effective and instructionally sound computer courseware. Try incorporating courseware into your lesson plans and see how enthusiastically your students respond!

Note: The courseware described above is widely accepted and has been used by the author in inservice training workshops for mathematics teachers—however, mention of a courseware product in this article does *not* imply endorsement of it by the author or by any organization. **END■**

[Lois Edwards, Instructional Coordinator, Minnesota Educational Computing Corporation, 3490 Lexington Avenue North, St. Paul, MN 55112.]

© Copyright 1985 ICCE

Computers in Science Education

Richard C. Adams

For Heaven's Sake

Astronomy is one of those subjects that doesn't have a lot of class; its place in high schools is usually not too strong. Oh, I know there are wonderful schools out there with 15,000 students who teach Astronomy I, Astronomy II, College Level Astronomy and Astrophysics Seminar. But my high school has 485 students in it and is considered a medium-sized school in Oregon. I kind of think there are a lot of schools of that size or smaller out in the hinterland.

Astronomy is one of those subjects that may be taken up descriptively in junior high school science as a special topic in a minicourse, or perhaps in senior high school as a unit in earth science or general science. Occasionally, it sneaks into physics, but usually only if the teacher has had some background in it.

In my own case, I had taken both undergraduate and graduate college-level astronomy courses, and with my hobby of stargazing (much more fun than pure mathematical astronomy), I wanted to share this interest with my students. One of my students got so turned on to the stars ten years ago that he now works at the Very Large Telescope Array in Socorro, New Mexico, in radioastronomy. Things like that can keep a teacher going for several Mondays at least.

The point of all this is that, although astronomy can have intrinsic attraction, to present it properly requires some expertise in the sources. Giving an interested student the line, "Here's a book. . ." may quickly short-circuit budding curiosity. However, this is a column about computers in science, and I have a few solutions.

THE ASTRONOMY DISK
By Sheridan Simon
Prentice-Hall, Inc., Englewood Cliffs, NJ 07632
$29.95
Disk for Apple II series
Documentation

About two years ago I was doing some beta testing of computer programs for a local software company. Sheridan Simon had submitted two disks, one on physics and one on astronomy. The local software company delayed too long in replying to Dr. Simon, however, and Prentice-Hall got the first bid.

Dr. Simon is not only erudite and precise in his science, but he also has a sense of humor that comes through strongly on the disk. Some of his explanations, while sticking mainly to the point, occasionally go off on little flights of clever remarks and some wild allusions. The opening graphics are a start, but the rest of the humor is in the reading.

I was so impressed with his writing style that I got him to share some of his thoughts in an article in TCT ("Unprintable Physics," *The Computing Teacher*, Vol. 10 #7, March 1983, p. 14).

"Designing solar systems, planets and stars, and then letting the whole thing run for a few thousand years is just not in the budget of most science departments this side of heaven."

The Astronomy Disk is not the latest in graphics. Dr. Simon has chosen to use graphics only in those circumstances where they enhance the program and can help to get the point across.

Broken up into three groups, "Space Travel," "The Solar System" and "The Stars," there are 16 separate programs on the disk.

"Space Travel"

Many teenagers have watched all the films and know that spacecraft and "X-wing fighters" zoom around space with dives, curves and hard banks. With the aim of coming a bit closer to reality, "Space Travel" uses graphics to allow students to try orbits of satellites, design a multistage rocket, figure out the starting points of a voyage to Mars (where both origin and destination are moving in their own orbits), and even see how constant acceleration changes the nature of time on long flights. Programs like the Mars flight have no calculations, but you have to choose a particular moment to fire the rocket's engine. Students who tried this needed about 10 minutes to get a feel for orbit-to-orbit plotting. The mathematics can come later, after they have an urge to know more.

"The Solar System"

"The Solar System" involves visual representations of the satellites of Jupiter, planetary/satellite systems you design, the motions of comets and an orrery that shows the planetary positions, as seen from any angle, for any year between 1500 and 2500.

"The Stars"

"The Stars" again uses graphics to show things hard to present in a book. You can view a galaxy from any angle, see how a star changes on the interior, build up graphs of radiation output depending on temperature or spectral class —and even build a world. This last program was one of the most popular with my students, but I feel an important part is now missing. Given choices of stellar class, stellar type, size of planet, albedo, mass of planet and distance from the star, the program figures out what the year and temperature would be. Students played with adjusting orbits slightly or adding reflectivity (albedo) to the atmosphere and the like, trying to get a terrestrial environment, but in wild solar systems at impossible distances. At that point the original program used to then draw a scene from the planet, showing how big the star would look in the sky, the size of mountains expected with that gravity, and snow, green grass, fried rock or molten surfaces, depending on temperature. My students liked this part the best because they could then stand on their planet and look out the window. I miss the fact that the new program has no such graphics. Perhaps Dr. Simon can be enticed to put them back in and remove something else.

Sheridan Simon has used these programs in his college level classes at Guilford College, Greensboro, North Carolina, as assignments done without teacher assistance. To really get the most out of these programs, about one hour should be spent trying out the variations and fiddling with the parameters. Thus, with 16 programs, you can look forward to about a week's worth of work and a

© Copyright 1985 ICCE

good minicourse in astronomy.

This is one of those cases where the power of the computer can be used to simulate a lab that would be very difficult for most schools to set up. Designing solar systems, planets and stars, and then letting the whole thing run for a few thousand years is just not in the budget of most science departments this side of heaven.

TELLSTAR: YOUR COMPUTER GRAPHICS WINDOW TO THE UNIVERSE

Scharf Software Systems, Inc., Suite 1068, 2111-M 30th St., Boulder, CO 80301

$79.95 Apple; $99.95 IBM; $3.00 additional shipping and handling charge

Apple II series disk (also for IBM)

Tellstar is something else. It is not meant to be a tutorial in any sense of the word. Instead, it is a tool for stargazers. True astronomers never get to look at stars; most of the time they look at strips of film showing spectra of stars to see minute changes in distance, speed or rotation. In fifth through twelfth grade-type astronomy, you need to take the class to see stars at least once. Here in western Oregon, I've had to pick my times very carefully to see any sky at all during the winter.

What *Tellstar* does is give you your own private planetarium. It's not as good as seeing the big one with all the lights out and the kids going "Ahhh!" However, it *can* prepare students for their field trip or what to look for each night, even if the skies maintain a constant drip. Using this program, you can see what the night sky will show for any location, any date from A.D. 0 to 3000, any time of day (it will even show the sky as it would look during the daytime, complete with sun), and looking any of eight directions to the horizon or overhead.

Stars, planets, Messier objects and constellations are shown. Moving cross hairs to any object will result in a full identification, including name, phase (planets), magnitude, location in right ascension and declination, time of rising and setting on that particular date and headings. You can move forward or backward through the calendar or the clock to see planets "wander" (as the Greeks thought in their name, "planet"), or see the constellations change with the seasons. Opportunities are available to print out the data on any object or even the sky map.

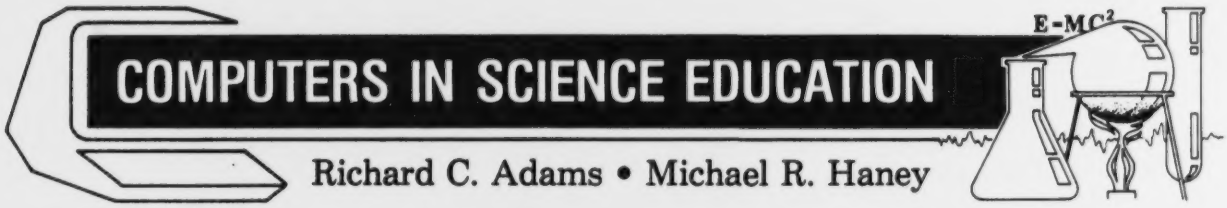
The program supports the graphics capabilities of the Apple Silentype printer (I personally don't know anyone who has one of those, of the 230 people I know who have Apples), but you can save the sky pictures on any initialized disk for later printing using the graphics dumps for *your* printer.

The 70-page manual contains numerous appendices with data on stars, Messier objects, planets and longitude and latitude readings for 63 North American cities and 39 cities in the rest of the world.

This disk probably won't teach kids as much astronomy as the one by Dr. Simon. It will, however, permit students to derive some things by tinkering with time and will allow the teacher to generate interest in field trips to the sky. ■

[Richard C. Adams, Pleasant Hill High School, 36386 Highway 58, Pleasant Hill, OR 97455.]

© Copyright 1985 ICCE



Doing Science

by
Richard C. Adams

During the year, I, as the reviewer for the Science column, receive software to review. A few months ago I stated I was interested in people using computers to **do science**, rather than telling about it with "automatic page turners." The following represent noteworthy examples of "doing science" software.

YOUNG PEOPLE'S MEDICAL SKOOL
Simulation Software Co.
6035 N. Maplewood Ave.
Chicago, IL 60659
Grades 7-12

Apple II+ or IIe
$27.95 (shipping included)

This is one of those programs that you suspect will be terrible. Is there a glossy brochure? No, my copy arrived with original printouts from a dot-matrix printer. Then there's the name—"skool." Really, now. Does the disk have a four-color label? No, it almost looks typed. The disk isn't even copy protected, so you can make back-ups before you use it in a hazardous environment (the normal classroom). What can you possibly expect from something as poorly presented as this?

What you get is a rather nicely done simulation of medical diagnosis. I was pleasantly surprised by how well the graphics were done and even the inclusion of spoken-word sound effects. I've tried out this disk with three biology teachers, one of whom teaches Health Occupations, and ten students; we all had a lot of fun with it.

Here's how it works. You get to be the doctor and are presented with some symptoms. Some of these are in the form of notes from a patient's medical history, some are by graphical displays of the symptoms such as bleeding or rashes on the screen (in a tender location), and some are from words spoken through the Apple speaker.

You then select further lab tests you may wish to do from a menu. You can keep doing lab tests until you arrive at a diagnosis or the patient runs out of money. (The bank balance is prominently displayed.) If you do too many tests, the patient occasionally gets disgusted and leaves. S/he may also die before you figure everything out.

If your treatment involves surgery, you are given a choice of anesthetic and how much to administer. Vital signs are read out and most actions are displayed as rather good pictures on the screen with a lot of clever shadings (giving about 100 different colors on the normally 7-color Apple graphics). In one case, I suspected appendicitis and had the abdominal cavity laid open at which time I was given my choice of which organ to remove. After suturing, a progress report was given.

You can run into difficulty and your cure may not work. You then have to keep trying other things. Occasionally you'll lose one of your electronic patients. A biology teacher and I were trying to cure a male patient of a condition where his vital signs kept going down, no matter how we varied his diet and tried other treatments. We were so interested in the next move that the other teacher leaned forward to make a drastic change when her foot knocked the computer power cord out of the wall. All life-support systems were then shut down and we'll never know. . . .

It's fun to work on a medical "detective story"—and you also learn a lot. A list of medical terms is given in the somewhat skimpy documentation, but you are encouraged to use a library for more help.

There is a Part II for sale at about the same price ($27.95) and an easier version for younger children at $42.25—neither of which I have reviewed.

EXPERIMENTS IN HUMAN PHYSIOLOGY
(Catalog #502-00-SASA)
HRM Software, a division of Human Relations Media
175 Tompkins Avenue
Pleasantville, NY 10570
Grades 9-12

Apple II+ or IIe
$249.00 with 30-day refund option

That price of $249 is kind of budget-straining when you first look at it. On the other hand, for that

© Copyright 1983 ICCE

price you get one disk, two temperature probes, an interface box, a timer plug, a light and sensor attachment, four extender cables, ribbon cable and a 75-page manual. (A back-up copy can be made.)

What does it do? The manual contains ten experiments dealing with human physiology. These are not *simulations* of human measurements; rather, you do the measurements on a human subject and then interpret the results. Reaction times of various parts of your body can be measured and temperature differences of different parts can be taken. Even your heart rate can be measured in a quite clever manner, with a light shining through the skin and a sensor checking to see how much light is transmitted or reflected, depending on how much blood is going by that point at any one time. The screen shows both immediate readings and long-period readings of the rate, so biofeedback techniques can be developed and slower responses also charted.

All of the probes attach by color-coded cables to an interface box with eight probe sockets. The box is plugged into the game paddle port inside the computer, but an extra game paddle port is on the box so you don't have to unplug everything if you want to use the port for something else in the meantime.

Using the equipment, you can even measure breathing rate by putting the temperature probe near the nose of a subject and noting the change in temperature on exhalation and inhalation. You can rig up a polygraph for "lie detection" should the need arise. Athletic departments can use the devices for conditioning measurements quite easily. The chemistry and physics departments can use the temperature and light probes rather nicely also, making the $249 price less costly if split among four or five departments who take turns using the equipment.

A demonstration disk is available which basically shows menu layouts and the like, with the response-time experiment actually included, since it does not require any of the hardware. Response time is measured after a stimulus of a sound or a flash on the screen.

I have not seen the whole package, but the demonstration is impressive. I conferred with a colleague who had purchased the package, and he had had good results with it and found very few problems. It *is* a lot of money (at least for my budget), but as I said, perhaps some sharing can be done. The manual should get you started with a number of intrinsically interesting experiments, but the package seems to encourage you to develop more uses, once you have precision equipment available. You may even discover that one shared package is not adequate, once you find out all you can do with it.

Producer's Reply:

We appreciate the opportunity to respond to the review.

The demonstration disk is available free on a 30-day preview.

Also available from HRM Software:

- *Experiments in Science,* a series of 12 experiments in biology, physiology, chemistry, physics and earth and planetary science. Hardware is also included in this package.
- *Health Awareness Games,* a set of 5 programs that draws on extensive tables of statistics on people's health.

MICROCOMPUTER PHYSICS LAB—HEAT
Cross Educational Software
P.O. Box 1536
Ruston, Louisiana 71270
Grades 10-12

Apple II + or IIe
$60.00

On a slightly lower scale, in both price and execution, we have the programs from Cross Educational Software. Dr. Mark Cross has written articles about educational philosophy in *Call A.P.P.L.E.* and elsewhere and has a firm foundation in physics as well.

Rather than an all-inclusive program dealing with many measurements, Cross has taken the method of providing modules. The one I reviewed was dealing with heat. There is another one dealing with light which measures the time interval between flashes of light or something blocking the probe, such as a cart moving down an inclined plane or a pendulum. For $60.00, you get a diskette, a three-ring binder for documentation, and the necessary hardware for four temperature probes. To use them, however, you have to put them together, which is not really that hard. It involves soldering four wires onto a 16-pin plug and the other ends to two thermistors (which look like little blobs of metal with two wires coming out). Everything is provided except the solder and a soldering pencil.

Rather straightforward and simple, the programs on the disk allow you to calibrate the thermistors (each one will be slightly different when you get them hooked up), get a continuous readout of temperatures on the screen, or, most importantly, see the temperatures displayed both in digits and in a nicely done graph on the screen. The range of the temperature graph can be set as narrow or wide as you want, and the time interval between temperature readings is adjustable between one and 1000 seconds. That means a graph can display 270 seconds (4.5 minutes) or as long as 270,000 seconds (75 hours). You could, for example, set up a heated object in a box with some specific insulating material and come back the next day (or three days later) and see how the temperature had dropped. A second thermistor could be reading the room temperature's variation at the same time and a third just be a control in a box of insulation. (A fourth thermistor is provided if you want to hook it up.)

© Copyright 1985 ICCE

Your graph would show all three lines in a quite readable form on the screen. If you wish to have a hard copy of the numbers, just turn the printer on before you leave and the numbers will be listed there instead.

The program could be easily modified to write to a file on the disk, although timing might be changed slightly.

You can do that sort of thing because Cross Software has no "protections" on the disk. You can copy it or modify it as you see fit. This program can easily be used by a student without much preparation, but full usage of this package assumes there is a real teacher out there who is willing to experiment with new applications. The manual gives outlines of five experiments for use and also includes listings of all programs used on the disk, including the assembly language timing routine, for you to learn from and modify.

I had a problem with the program when I first received it. I wrote a letter to Dr. Cross and had a new, corrected version back in the mail within five days. A letter accompanied it that showed I was dealing with a real, live person who cared about how his software worked for teachers.

Again, although this software does only basically one thing, it does it well and at a reasonable price.

Also available from Cross Educational Software: 12-volume *Physics* sets ($260.00 total or $15-$35 each); and *Physics Lab-Light* and *Physics Lab-Sound*, both at $60.00.

PRECISION TIMER $39.95
GRAPHICAL ANALYSIS $24.95
RAY TRACER $24.95
Vernier Software
2920 S.W. 89th Street
Portland, Oregon 97225
Grades 10-12

Apple II + or IIe

David L. Vernier has taken the "module" approach, similar to that of Dr. Cross. In this case, however, the equipment does not come with the software.

The disk called *Precision Timer* is basically a timing device to tell when one of the paddle buttons has been pushed or released. In the operation of this disk, the activation is by light, rather than by pushing a button, although it could be modified to do that.

The disk is designed to operate with "photogates" in a module from PASCO (part #ME-9204). Each of those gates costs $75.00 and you will use two of them. An interface box to the game port costs $100.00 (part #AI-6575). On the other hand, you can build the interface box with *three* photogates using parts bought at Radio Shack for $24.12. Mr. Vernier gives instructions on what parts to use and how to put them together. I know how to solder but am weak on semiconductor devices. Even so, I was able to put together the whole thing in about three hours. The worst part was going around to five Radio Shack stores to get the 18 parts, since each store seems to only have one of each item. Phototransistor FPT-100 usually consists of a card on a rack with "Sorry, out of stock" right behind it. Other than that, putting the hardware together is fairly straightforward.

Once you have everything together, the software does the rest. There are 12 different modes of measurement, depending on how many light sensors you wish to use. Basically, each works by cutting off light to a very small sensor. This begins the timing, and then turning off another or the same one again stops the timing. You can drop a ball past two sensors and get the speed, past three and get the acceleration, let a pendulum go past one sensor twice and get the period, perform collisions with carts or airtracks and get all the data you need. You can also use the keyboard, without any other equipment, to make this program a quite accurate stopwatch (+ or − 0.0002 seconds).

One nice feature of the disk is that all numerical displays can be shown on the screen in numbers four times as big as the normal digits on your monitor. Thus the data collection can be seen from the back of the room for demonstrations or larger group labs.

One caution about this equipment: the third light sensor uses paddle button #2, which is also the pin used in most "shift key modifications" for the Apple. That means that it will appear that the shift key is being held down. One way to do the shift key mod is from the #1 pin of the chip just in front (toward the keyboard) of the game paddle port, to the keyboard #2 pin using spring clips (Radio Shack Cat. #270-372). Installation takes about 15 seconds, but more importantly for this experiment, *disconnecting* it can be done in about 5 seconds.

Up to 100 data points can be collected, and then average and variance will be printed out. For many of the experiments, further mathematical treatment is possible, specific for the usual use. For example, the pendulum experiments will give the data of period, frequency, etc. that you would normally calculate.

The 42-page manual contains a user's guide, a teacher's guide, modifications to the equipment and programs that can be made, and a complete listing of all programs with a table of variables. The disk is completely unprotected so back-up copies and modifications are easily made.

Graphical Analysis can interface with the data from the above experiments, collecting stored data from the disk directly. Or, you can enter pairs of data for up to 32 points.

The disk is a very quick way to visualize data. Putting in the points is one option. Using the Plot option, you can label the axes, put in the units to be displayed, plot just points, points with connection,

© Copyright 1985 ICCE

or "point protectors" of boxes around each point. A linear regression analysis can plot the closest fit straight line also. Other menu options give you the choice of exchanging the axes, changing any pair of points, and a numerical analysis section giving slope and intercepts.

The program doesn't do anything really fancy, but for $24.95 you get a very quick way to display lab data in a graphical form. The scales on the side can be done in "auto-scale" mode for quickness, or you can determine the value of each unit for clarity. Although this is not as instructive as learning how to make a neat graph with ruler and pencil, perhaps your interest is in teaching how to *set up* graphical displays and what they *mean*, rather than the mechanics of pencil and paper neatness.

Ray Tracer is an electronic blackboard. There is nothing on this disk that a good physics teacher couldn't do with a blackboard, chalk, straight-edge, protractor, compass, fantastic skill in drawing and three hours for each class period.

The key differences are in accuracy and time. I can draw a good picture of what a light ray will do at the interface between air and glass. But I can't do it accurately for both red and blue light, for any pair of materials, and for 15 different angles in under two minutes. I can *talk* about the "critical angle" where light bounces off of water, but it's more instructive to "sneak up" on it by changing the angle of incidence slowly until the reflection takes place on the screen.

Besides refractive indices between two media, other options from the menu of 15 situations deal with thin and thick lenses, compound lenses, plane and spherical mirrors. In each case, one white ray will be drawn, or you have the option of doing two rays (red and blue) to show the difference wave length makes. The programs can be run by students as a "game" since a source is picked arbitrarily and then a target is placed somewhere. The object is to hit the target, no matter what media you have to go through or reflect from or refract through. This sort of "gaming" can teach a lot of intuitive feelings for what optical devices do without a lot of difficult lab setups.

The 29-page manual has a user's guide, a teacher's guide, what optical laws are being demonstrated and a list of experiments and references. A complete listing of the programs and variables used are also included on this unprotected, LISTable disk. You are given hints on modification and are encouraged to do so, to make the programs fit your exact requirements.

If the drawings aren't fast enough for you, there is also a *compiled* version on the disk that runs 2-3 times faster, although modification or even listing is then impossible.

Again, at $24.95, this disk doesn't allow you to do anything you couldn't do before, except improve accuracy, save time, and make optics more interesting. Perhaps such things are attractive to you.

Also available from Vernier, but not reviewed: *Orbit, Projectiles, Kinematics, Charged Particles, Vector Addition,* and *Wave Addition,* all at $24.95. (These programs have also undergone recent revision.)

Author's Reply:

Graphical Analysis has undergone major revisions with several new features added for the same price.

- 200 pairs of data may now be entered, instead of 32.
- Data is rounded before it is displayed—the number of significant figures is controlled by the operator.
- The appearance of the final graph has been improved.
- Semi-log and log-log graphs may be displayed.
- A hard copy of the graph can be made on any standard 80-column printer.

END■

© Copyright 1985 ICCE

"I did a good job on nouns and verbs today. Could I try adjectives tomorrow?"

"We need another new dictionary, Mrs. Jacobi. This one got worn out!"

"Billy is hogging the computer again!"

"Boy is this a neat piece of software! Can I come back and finish it after school?"

"Mrs. Jacobi, I just LOVE computers!"

by
Carolyn Jacobi

In July of 1982 a three-year Title VII demonstration grant was awarded to the Rapid City School District. It was the largest grant awarded that year, and its reason for existence was to determine whether or not computer assisted instruction could help Native American children. Two and one-half years later, we think we have some answers.

Statistically Speaking

by
Carl Edeburn

My experience with this project has been truly rewarding. As a teacher-educator I have been involved in and intrigued with the potential of CAI since the early 1970s. However, experience in research and evaluation has caused me to look with some skepticism upon the blind rush to infuse computers into all aspects of education without a collaborative effort to measure the impact. Working with this project has allowed me to "scratch the surface" in establishing a better understanding of the potential computers have for improving the educational environment.

The ultimate goal of this three-year project was to improve the reading and language skills of an identified group of Limited English Proficient Native American students.

Lakota (Sioux Indian) children constitute 13 percent of the district school population. In the first year of the project, efforts were focused in two elementary schools which contained the highest numbers of Lakota children. These included a school with a 55 percent Lakota population and a second site which had a 38 percent total. In Year II an additional site (16 percent Lakota) was selected for participation.

Since the use of such a technologically innovative approach was new in 1981, there was very little research in this area. Hence, the project was plowing new ground in the areas of curriculum development and pedagogy and had implications for baseline research as well as the fulfillment of the immediate educational goals. The purposes were:

1. To develop and incorporate into purchased educational programs instructionally appropriate materials;
2. To develop pedagogically appropriate programmed materials for use with microcomputers; and
3. To train instructional staff in the methods of CAI technology.

The students were chosen using a combination of methods including standardized test scores, assessment scale scores,

© Copyright 1985 ICCE

and a review of demographic information which included family income and teacher/principal input. A total of 108 students were involved in the project.

A dedicated, capable Computer Specialist is a must in a project of this type. Without Carolyn Jacobi's expertise as a teacher and her interest in both the practical and research elements of this project, my participation would have been less meaningful and the project would certainly not have been as successful as the findings reveal.

Although the project was monitored via several data sources in the evaluation process, my discussion is focused on pre- and post-test results in the Comprehensive Test of Basic Skills (CTBS) and the CAI activity file.

The CTBS subscale scores for Word Attack, Vocabulary, Comprehension and Composite Reading, as well as the score for Spelling, were retrieved from the fall and spring testing process to determine whether or not growth had occurred.

To address these dimensions, the Norm-referenced evaluation model (N-RM) was employed. In the N-RM, performance (or gains) of the treatment group is compared to the performance of the norming sample, i.e., the group from whose performance the test forms were derived. Therefore, the norming sample (national norms) are used as a comparison group against which to compare the growth or gains in the program group(s).

Fall and spring tests were administered within two weeks of when the norms were obtained. The norm-referenced comparison was implemented on the individual and composite reading scores, as well as on the spelling scores of the CTBS.

In the second year of the project, 11 (85 percent) of the 13 subscale by-grade comparisons evidenced significant growth, and all of the total group comparisons revealed significant growth with probability levels of < .001. The total group comparison is summarized in the following table:

Summary of Means, Mean Growth, t Values and One-Tailed Probability Levels by Subscale by All Grades
N = 108

Subscale	Fall NCE Mean	Spring NCE Mean	Mean Growth	t	P
Word Attack*	35.95	51.59	15.64	4.64	.000
Vocabulary	35.87	45.56	9.69	7.48	.000
Comprehension	37.09	44.31	7.21	5.26	.000
Composite	36.51	45.33	8.82	8.05	.000
Spelling	39.18	46.33	7.16	5.44	.000

*Grade 3 only (N = 22)

Background

After many years of "Indian programs" in the local school system, the Native American student was making no significant progress when it came time to interpret test scores each spring. This bothered Dolores Riley, then Title VII Director. Would another approach prove to be more successful? And, at the same time, would another approach provide these Native American children with in-roads into computer use? (Computer access at that time was limited primarily to students in schools where PTA funds or parents' clout could underwrite the school budget.)

Rapid City is a town of 65,000 people, located at the foot of the Black Hills. Mount Rushmore is 30 miles south and Bear Butte, the holy and sacred mountain of the Cheyenne and Sioux nations, is located 30 miles to the north. There are nine Indian reservations in the state. The reservation is the psychological and social/cultural center of the lives of many of our students. Although families often move to Rapid City to find work or attend school, weekends and vacations are spent on the reservation. And many students actually move back and forth from reservation to city several times in the course of the year.

Carolyn Jacobi, Computer Specialist; James Livermont, Title VII Director; and Lowell Amiotte, Director of Indian Education, greet newest staff member, TOPO.

In 1982 our district offered programming classes at the high school level, a Computer Literacy Committee was investigating the role of computers in the district, Chapter I personnel were using drill and practice tapes, and some enterprising teachers were bringing their own computers to their classrooms. In 1982 the idea of using computers in a lab as an integral part of some elementary students' instruction was an innovative idea indeed.

A needs assessment done in the target schools and in the community determined that area parents and teachers were most concerned with reading and language arts scores. Since aides were already working with K-2 teachers in the impacted schools, the grant was written for students in grades three through six. During the summer of 1982, introductory computer activities were planned, a computer specialist was hired, and workshops were held for teachers in the two most heavily impacted schools, where the first two labs would be placed. Computer aides were given two full weeks of training, and a "kick-off" day was held in early September for the staffs of both schools. In nearly every case, these workshops were the first computer experiences for either teachers or aides, and they were extremely valuable.

The Program

The labs operate similarly to each other, but there are some important differences. The manner in which children are tracked, their time on computers, and the types of software they use is very exact. Another constant is the make-up of each group of children brought into the lab. All groups have 60 percent identified students and 40

© Copyright 1985 ICCE

percent non-identified. This has had a positive influence on the success of the labs, because it has kept project students from being identified, either with a special minority program or with a remedial program. In fact, it has given us the ability to give time on the computer to many children, since most classroom teachers used the 40 percent of their allotted groups to rotate their other students through the lab. This allowed us to be instrumental in the computer literacy process of students other than the ones we served, and it gave our identified students the prestige of teaching "new" students in their class.

An area of difference in operation has to do with the way the children are scheduled. Working with the building staffs, it was decided to pull students out of class for this program, even though pull-out programs normally have a low degree of acceptance in this district. By tailoring students' time out of class to the teacher's schedule, the degree of negative response was cut considerably. Some staff schedule children on a rotating basis, some on a fixed time allotment—and one school has a mix of the two. Picking up students from their classrooms and delivering them on time makes the program more acceptable, also.

Students are in the program for a half hour each day. One day a week they are in a tracking/testing system, with aides keeping a careful watch for student needs and further work to plug in software practice. Three days a week students are given other pieces of tracked software and, if attendance and work are satisfactory, they have their choice of software on the fifth day.

The size of the t value is an indicator of the difference between the fall distribution of NCE scores and the spring distribution of NCE scores.

The P score is an indicator of how many times this difference (a t value of this magnitude) would be expected by chance alone in 1,000 replications of the study. As can be noted in the table, all of the differences (mean growth) were of sufficient magnitude to generate very small probability levels. Common sense would demand that in the case of the greater difference (t) and the smaller probability (P), the more convinced we would be of any real or actual changes.

The evaluation process incorporated a dimension which attempted to pinpoint those elements of CAI participation (treatment) which had the greatest effect upon growth in the three major reading scores (Vocabulary, Comprehension and Composite). The approach employed was stepwise regression, a statistical approach which is based upon the principle of correlation or co-variability between and among variables. These variables were the time each child spent interacting with the 10 different software packages employed in the project (predictors) and individual NCE gain (criterion) on the three subscale scores.

An interesting pattern emerged from these three analyses. Although three or four statistically significant software packages emerged from each of the regressions, one package emerged as the single best predictor in all three instances. This software package, *Microzine*, had characteristics which included programming, spatial relations, graphics (Logo-like) activities, and organizational sequencing, all of which are problem-solving "proactive" learning activities as opposed to the typical "reactive" CAI drill and practice routines.

As a part of the evaluation process, by-sex statistical comparisons were implemented. These analyses revealed no differences in the NCE gain scores of the students when compared by sex.

Many educators have indicated that there is substantial inequity among schools. According to Stone (*Teaching and Computers*, January 1985) and *The Computing Teacher*'s Equity issue, April 1984, schools in high income areas are characterized by the following:

1. More money.
2. Parental support of the technology (hardware, software).
3. Community support for teacher training and inservice efforts.
4. Students learn to direct the com-

© Copyright 1985 ICCE

puter through programming, data base management and word processing.

On the other hand, schools in low income areas are characterized by:

1. Less money for computers.
2. A constituency of parents who are either not educated or able to express their concerns about current educational issues.
3. Student use of computers (if they use them) largely restricted to drill and practice.

Stone indicated that the CAI approach in low-income schools is a result of the amount of teacher inservice. The implementation of drill and practice and other reactive software activities requires much less teacher inservice than do processes such as programming and problem solving.

An interesting parallel to this argument is the reactive approach to teaching writing in the language arts, where many schools rely upon worksheets or workbooks in the areas of structure, grammar, usage and punctuation in the hope that students will somehow "put it all together" when they attempt to write something. Other schools spend the time actually teaching processes of communication, including organization, exposition, and actual problem solving. *Instruction* in writing is much more demanding than processing students through a grammar workbook, and hence is more expensive.

This study tends to corroborate the value of software problem solving and inductive CAI activity in that they contribute positively to growth in both vocabulary and comprehension, whereas some of the packages, characterized by drill-and-practice routines, were actually negatively correlated with the NCE gains.

This finding may well make us question the efficacy of typical drill-and-practice software, and focus instead on incorporating problem-solving activities in the teaching of reading skills. In the third project year we are attempting to focus on problem-solving activities.

END ■

[Dr. Carl Edeburn, Professor, Division of Education, Box 2220, South Dakota State University, Brookings, SD 57007.]

Finding language arts CAI software was no easy task in 1982. We used what was available. We found a tracking system, *PAL* (Universal Systems of Education), which contained a detailed, comprehensive breakdown of objectives that satisfied the most structured language arts teacher. This tracking system also provided a printout which was given to each child at the end of each week to take to his/her teacher and family. The need for this exacting communication is no longer necessary, but initially it helped to establish program credibility. This same system is still in use—nearly the only one that is—but the reports now include only a weekly printout for our files. Teachers and parents are given one on request, and a copy is submitted routinely every nine weeks for inclusion in the language arts report card.

In addition to the careful documentation of the curriculum, countless hours were spent talking with individual staff members, parent groups and interested community members. A software review committee was established, made up of parents and teachers. This is no longer needed, either. Our curriculum is trusted. Many teachers in the four target schools request certain pieces of software to be purchased or used with students when a specific problem is discerned. In fact, a bilingual program has been requested in more schools than we can cover—a first for our district.

The Results

Test scores after the first six months of operation were "statistically significant" in one school and "educationally significant" in the second. In 1983 we added a third laboratory, and this past year we attained "statistically significant" growth overall. It is also significant that we have observed no sex-related differences in adaptation to and/or liking of the computer. The vast majority of both boys and girls like the computer and benefit from its use.

The happiest part of the entire experience is that it tends to confirm what good teachers suspected all along. We have always said (haven't we?) that children need to practice reading more to become proficient at it. Records of time spent on software packages measured against growth revealed that ***the software packages that appear to make the significant differences are those that deal with imaginative reading and decision making . . . LOTS of reading and decision making.*** I was delighted!

The software below (in the order listed) had the greatest impact on our students. The absolute best was *Microzine* from Scholastic. It gave students lots of reading practice, and they liked it a lot! After that was:

- *Gertrude's Puzzles*, *Rocky's Boots* — The Learning Company
- *PIK-PEK-PUT*, *Riddle Me This* — Data Command
- *Weird and Wacky World I and II*, *Dragon Game Series* — Educational Activities

We used two other packages which I can't recommend, both because they showed little correlation with students' growth and because they were terribly expensive.

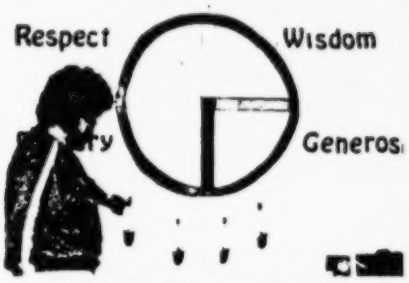
© Copyright 1985 ICCE

We also used a variety of other non-tracked software:

Code Quest (Sunburst)
Jigsaw (Island Software)
Arcademics (DLM)
Moptown Hotel (Apple)
In Search Of. . . (Spinnaker)
Some MECC disks.

The labs exchange software every nine weeks to keep material fresh.

The thesauruses and dictionaries really get a work-out!



Elaine Delores places display of the turtle's playground, Indian-style.

In the beginning, the lack of good educational software was the main problem. We moved out of the drill-and-practice mode as soon as possible and into tutorial packages. The software market is constantly monitored and more creative packages used whenever they fit the outline of the grant. Now the constrictions of the grant are the biggest source of frustration. Of late, the need to do more than CAI is evident, but we have been unable to move into the programming aspects of computing, and have used word processing only as a special, supplemental part of the program, even though nearly all of the aides have been trained in the teaching of writing, word processing and Logo. Many of the students now program in low-resolution graphics on the Apple. The combining of graphics and word processing with original stories of children just beg to be done and we are all excited with the possibilities, but the grant reads that each child will receive 30 minutes of computer time each day in language arts instruction, and we have adhered to that very strictly, documenting daily records of each student.

One "teacher pleaser" that I am happy to pass along is this: When you budget, be sure to include money for an ample supply of dictionaries and thesauruses. The amount of use they receive surpasses our wildest anticipation. The labs are given two copies of each at the beginning of the school year and these often must be replaced well before the end of the school year. This use of reference materials also speaks highly of the aides and their ability to know when to help and when to encourage student independence. The quality of personnel continues to be the main reason for success of this project, and program personnel are given many opportunities for recognition.

One area that required a lot of time during the first year of operation was the matching of paper and pencil tasks to the objectives being taught on the screen. The assumption was made that this was a necessary follow-up to the "real world" classroom situation. After the first year's analysis was made, the most significant factor to growth was the actual time spent on the computer. The paper and pencil activity portion was dropped the second year, and growth was even more significant the second year. We now feel safe in saying that our time and energy can be more beneficially used on the computer, rather than on doing paper and pencil match-ups.

The Lakota (Sioux) culture is in no danger of "technology takeover" in our laboratory schools. Each of the computer aides is Native American, and a concerted effort is made to keep the Indian culture very visible in the teaching materials and in the lab decor. Our logo, Iya, is taken from an old Sioux legend, and we are now anticipating making our TOPO robot speak Lakota.

Since the jury remains out on the keyboarding issue, we address it in a practical manner, i.e., we ask only that students use both hands on the keyboard—we make no attempt to teach formal keyboarding. It has been interesting to watch some near-classic styles emerge!

We are nearing the end of our three-year demonstration grant period, which is now in its dissemination phase. Unlike many projects, which simply *end*, the recommendations from our administration to the local school board will be that the essentials of the labs be continued in pared-down form.

The experience of this project almost defies description. My wish would be that every classroom teacher would have the opportunity to be involved with an honest attempt at educational research. It hones ideas, feelings and readings into sharply defined objectives and educationally defendable goals in a way I had not thought possible. And, if the efforts are labeled very successful, the taste of the icing on that cake is unbelievably sweet!

"Mrs., Jacobi, I am calling from Stony Plain, Alberta. We just purchased our computers, and your name was given to use as someone who might be able to help us." "I'm looking for a good disk to buy for Tad. He has some problems with reading comprehension." "We are looking for a speaker. Are you TOPO the Robot's mother?"

It's nice to be able to help sometimes! **END■**

Norman Blue Arm, artist

[Carolyn Jacobi, Computer Specialist, Bilingual/Bicultural Education, Rapid City Schools, 21 St. Joe St., Rapid City, SD 57701.]

© Copyright 1985 ICCE

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

SOFTWARE REVIEWS

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Davie Moursund

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

GERTRUDE'S SECRETS
Reviewed by: David Ouellette, University of Oregon, Early Childhood Education Center
Producer: The Learning Company (TLC)
4370 Alpine Road
Portola Valley, CA 94025
Version: © 1982
Target Audience: Ages 4 to 9
Hardware Required: Apple II with one disk drive, color monitor (joystick optional).
Cost: $44.95

GERTRUDE'S SECRETS creates a wonderfully rich, simple and educationally sound "microworld" with places, objects and relationships that the young children I observed wanted to explore actively. As the children learned to move in Gertrude's world, to pick up objects, move them and to set them down, a magical door opens (SHAZAM!).

Inside Gertrude's domain the child constructs personal hunches and theories about possible relationships of shapes and colors in the puzzles which they test immediately as they play. I saw children joyfully incorporate correct hunches into what they knew about color, shapes and patterns. The children generated their own positive self-concepts about their own problem-solving abilities as they played and solved puzzles. Each good hunch can lead to another here, so Gertrude's world helps reinforce the value of working out one's own understandings, of trusting one's own learning abilities.

Gertrude's world holds three different kinds of attribute relationship puzzles. Each kind has a simple form and at least one more complex form farther out from Gertrude's central nest. The microworld is completed by a treasure room which stores the treasures a child accumulates by solving puzzles and by three other rooms for playing piece shapes and shape changing.

While the strengths of GERTRUDE'S SECRETS far outweigh its weaknesses, I found that colors of closely packed playing pieces misleadingly change color as they bump into each other. On the other hand, GERTRUDE'S SECRETS is the most appropriately implemented microworld (see Papert's *Mindstorms*) that I have seen for young children's learning about recognizing and using logical relationships.

Specific strengths of GERTRUDE'S SECRETS include:

- its simplicity for non-readers, especially the input routines
- child-controlled pacing, choice of movement and changes of intention
- an optional shortcut entrance into Gertrude's world for experienced players
- effective rewards for solving puzzles and for continuing to solve them
- tasteful (not brash, flashy or loud) use of sound and color
- the possibility of many different ways to solve any one puzzle
- the opportunity for an adult or researcher to observe a child's problem-solving strategies in an engaging and standard environment

GERTRUDE'S SECRETS is a gem, the kind of software new educational programs will be judged against. It presents young learners with a kind of logical relationships wading pool, safely enjoyable, a place to splash around, an effective environment for making a beginner's sense of the logical relationship waters they will spend a large part of their school years mastering.

Publisher's Reply:

In response to the reviewer's complaint about the color change of the playing pieces that bump into each other:

This is a hardware problem. We realize that figures that are close together on the screen of the Apple computer will meld their colors. Software designers should be aware of this problem. We were sorry that we were unable to solve it.

© Copyright 1985 ICCE

Gertrude's Puzzles (1982)
The Learning Company, 4370 Alpine Road, Portola Valley, CA 94025
Target Age Group: 6 and up
Hardware Required: Apple II+, Apple IIe (with caps lock down), color monitor
Cost: $44.95
Additional Requirements: Joystick optional
Reviewed by Johnny W. Lott, University of Montana

Gertrude's Puzzles is a set of puzzles designed to help children develop reasoning skills. In these puzzles, players learn how to sort and classify objects. They also learn to solve problems with incomplete information and must analyze what they see. The program consists of two main sections. The first is a tutorial section which teaches the user how to pick up and drop shapes, control speed, and do all the things they need to know to move around in the world. The second section is the game part. Players solve puzzles by using loops (Venn diagrams), networks and boxes (which use classifying and sorting in a manner reminiscent of magic squares).

In all puzzles, players move pieces to form color and shape patterns. Some of the puzzles give players rules for playing, while others force players to determine the rules. When a puzzle is solved, Gertrude brings a treasure.

There are six different puzzle rooms—a simple and more advanced room for each of the three types of puzzles. In the network puzzles, misplaced pieces move slightly if placed incorrectly. In the loop puzzles, pieces fall out of the puzzle. Examples of completed puzzles are given in rooms that do not have doors—only viewing windows. The initial shapes used are geometric shapes, and another room is the Shape-Edit Room where players may create their own shapes.

Strong points: The software is attractive with very good graphics. Directions are very clear for players who are reading at fourth grade level. Some skill is required to use the control key to move a key to open the door to start the puzzles. The I, J, K and M keys are required to move the cursor up, down, right and left, and the repeat key will make the movement faster. A player can take as much time as needed. This software can be used by a player with little teacher direction.

Weak points: The reading level is too advanced for most six-year-olds. Average fourth graders thoroughly enjoyed it, as did talented seventh graders. In the software the impression is given that a joystick must be used in the Joystick Room, but that is not true. In the Blue Room of direction, the control key and space bar will not let the orange square be moved as the directions state, and there is a misspelled word. When first using the software additional problems emerge, which include the cursor not always picking up the piece that it is on, but another that is very close. In the Box puzzle, one piece is under another piece at the start of play. This can be very confusing to beginning players.

Recommendations:

Gertrude's Puzzles is a fascinating set of logic puzzles for both children and adults. It is as effective, or more, than using attribute pieces to teach classifying and sorting. It is too bad that the reading level is too high for the advertised age group, but otherwise it is an excellent piece of software.

[Ed. note: We have seen a final draft of the revised manual, and want readers to know that future editions will include an even more elaborate and helpful manual.]

Publisher's Reply:

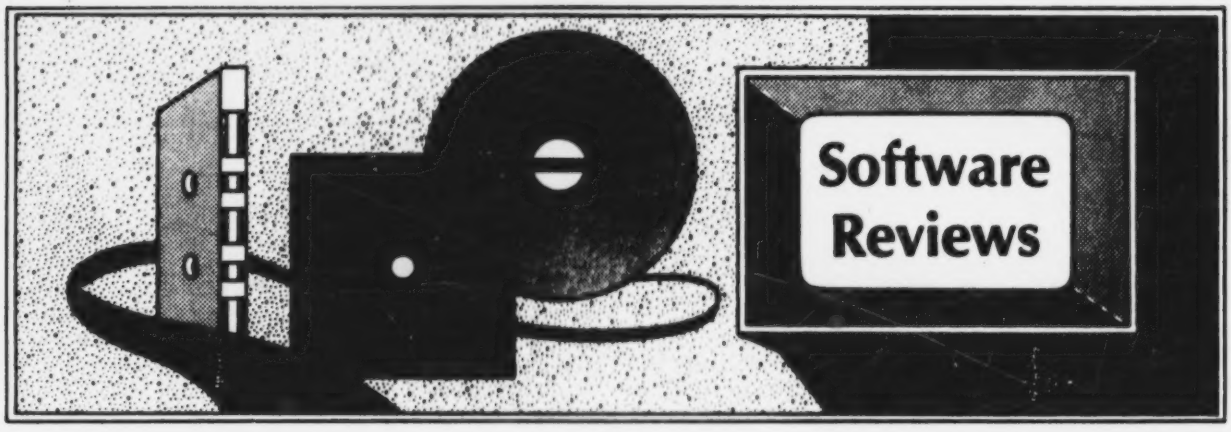
Thank you for sending the proofs for the upcoming review of *Gertrude's Puzzles* for TLC comment. The educators within our Research and Development group have evaluated the issues the reviewer raised and have provided the following responses.

1. *Reading level too high for average six year old*—The age range six and up means six and up. Not all six year olds can be expected to be at the same skill level; however, some will be able to understand the reading/instructions.
2. *Orange square doesn't move*—The colored part of the cursor must touch it in order for it to work; however, in the new version this is not required.
3. *Misspelled word*—This is also fixed in the new version.

Again, thank you for allowing us to comment.

Karen Roebuck
Public Relations
The Learning Company

© Copyright 1985 ICCE



Editor
Doris Carey

ROCKY'S BOOTS

Producer: The Learning Company. Available from local Apple dealers.
Reviewed on: Apple II Plus (48K), 1 disk drive, color monitor recommended.
Cost: $75
Age Level: 7 years and up.
Options: Joysticks.

Reviewed by Eric Anderson
Library/Media Services, Dakota C.U. Unit 201
Dakota, IL 61018

". . . Now somewhere in the black mountain hills of Dakota
There lived a young boy named Rocky Raccoon
And one day his woman ran off with another guy
Hit young Rocky in the eye Rocky didn't like that
He said I'm going to get that boy
So one day he walked into town
and . . ."[1]
went to a computer store
to buy himself a copy of ROCKY'S BOOTS.

The more software one reviews, the more cynical one becomes. Very rarely do programs excite me; the state of jumping up and down over a software item has passed. I remember the awe *DB Master* and *Screen Writer* and *Super Disk Copy* produced in me the first time I ever saw them run. *Castle Wolfenstein* caused me to give up 700 hours of my life before I escaped. It's been a long time since that sort of excitement returned. *Rocky's Boots* has rekindled that flame.

If the computer is indeed another piece of AV equipment, then librarians need to use the same criteria for microcomputer software selection that they use for other forms of educational media. A complement of sight, sound, color, motion and gee-whiz are the forte of the microcomputer. Many programs use one or more of these components. Never have I seen a software piece use all of these to the extent they are utilized in *Rocky's Boots*.

Rocky's Boots is a game designed to teach Boolean Logic and basic elements of computer circuitry to people seven years old and older. It uses a unique combination of hi-res graphics, sound and motion to fascinate the computer user into creating machines to perform tasks.

The game aspect is really underplayed. It took me three hours to realize that the object of the game was to score points. One scores points by making a machine to kick object targets out of a trough. The object targets vary in shape and color and the player must connect logic gates in such a matter that only the objects of a desired shape or color are kicked when they pass through sensors.

Most computer games are threatening. Here come the aliens—blow them out of the sky before they wipe you out. Succeed or die. *Rocky's Boots* does not play to this audience. Games also have predetermined time frames and limited sets of characters. You can stay with *Rocky's Boots* as long as you care to—succeed or not.

In *Rocky's Boots*, the user starts out by learning how to move and pick up pieces. Then one learns the different types of pieces to put together and the forms for construction. The player goes from room to room collecting parts to connect. Boolean logic is taught by using gates and electricity flow. Current will pass through an "and" gate connected to sensors. Or it will react in a "or" situation or a "not" situation in the same manner. The flow can be connected to delays or timers with the object of turning *Rocky's Boot* on to kick objects out of the target area.

Most computer games have a preset number of options. *Rocky's Boots* goes one better by providing games or letting the user change and modify the shapes or colors to create their own games. The user can also change the speed of the objects and halt them on demand. If that is not enough, one can change the point values of the targets. If my math were better I might be

[1]The lyrics from the Beatles song "Rocky Raccoon" are © 1968 Apple Music Co.

© Copyright 1985 ICCE

able to predict the number of probable games to be devised using *Rocky's Boots*. Thankfully I can't count that high.

The drawbacks are minimal, and I will attempt to come out of my euphoria to describe them to you.

Because the game is all hi-res, it takes a long time for the program to load from disk. When such utilities as Bill Basham's *Diversidos* are available, I can't see why companies don't utilize them to speed up loading.

I have always harped about price versus performance. $75 is a lot for a computer game. When I called The Learning Company asking them about volume discounts (I wanted to buy one copy of *Rocky's Boots* for each of the computers in my district—a total of 12), they told me there was no discount save the 10% educational discount. Readers should make a point of checking with individual dealers to see whether they offer volume or educational discounts.

There is the option of playing the game with joysticks. I do not own any, the district does not own any, and I much prefer to use the keyboard. I escaped from *Castle Wolfenstein* in 700 hours without them and *Rocky's Boots* performed well without them.

The use of color is very important. I played it for a long time on my green screen at home and had some difficulty. When I took it to school and used the color monitor it was much easier. With small children where the frustration consideration is very important, color monitors should be used with *Rocky's Boots*, at least at the more advanced stages.

One small point that my student programming ace Kent Welling brought up was the representation of electricity. He claims it moves much faster than displayed on the *Rocky's Boots* screen. When a switch is tripped or the cursor passes over a wire, current moves in a direction. The program animation of this is very slow. I guess that's what happens when you teach 17-year-olds the laws of physics.

The package is your standard Learning Company product. Disk-sized holder with thorough documentation booklet. As with all good software, I booted it first and played with it for three hours before I looked at the documentation. There is a nice full-sized page foldout detailing the world of "Rocky's Challenge," the main game on the disk. Let's see, I have a xeroxed copy of that map by my computer at home; one hanging above the computer in the library, one pasted inside my briefcase. . . . Do you suppose I am getting possessed?

It's very difficult to describe my reactions to *Rocky's Boots*. There is a "wow" sort of response that goes back to my first set of Lincoln Logs and Tinker Toys—building and fantasizing all wrapped together. Added to that is the thrill of putting my first radio set together and all of the really neat science labs I remember from 5th and 6th grade.

I have no conclusive reports detailing how well *Rocky's Boots* works with younger children. Some of the junior high and high school kids have fought over my copy of the disk. As of yet we have not gotten the disk into the hands of our primary people. My daughter has played it once (she's just turned six) and enjoyed it. Presently she is so wrapped up over Santa Claus and the holiday season that the computer isn't very exciting.

As long as I have been reviewing software packages I have screamed about price versus performance. A game costing $75 is without question an expensive proposition. *Rocky's Boots* is the first $75 educational software item I can recommend without reservation.

Publisher's Reply:

A comment on the representation of electricity: the program was designed so the user can see the direction of the flow. This is why the electricity seems to flow slowly. We are aware of this feature and it was done on purpose.

All our products are thoroughly field-tested in both home and school at a variety of age levels. We sometimes follow up on suggestions made by children and make changes.

Rocky's Boots is being tested at the Bing Nursery School at Stanford University with five- and six-year-old children. It is also in use at the Capitol Children's Museum, Harvard University and Columbia University. Graduate students at MIT are also using the program.

© Copyright 1985 ICCE

THE PRINT SHOP

Broderbund Software, 17 Paul Drive, San Rafael, CA 94903

Purpose: Creation of signs, banners, letterheads, etc.

Hardware Required: Apple II+, IIe, IIc and III in II-emulation mode, Apple DOS or ProDos, 48K of RAM and one disk drive.
Commodore 64 and one disk drive.
Dot matrix printer with graphics capabilities.
KoalaPad or joystick (optional).

Cost: $49.95 (Apple version); $44.95 (Commodore version)

Reviewed by: Merrianne Metzger, Computer Education Director, Bend-La Pine School District, Bend, OR

I predict that *The Print Shop* will be purchased by more schools than any other piece of software during the 1984-85 school year. Strong words, eh? But then this is a truly innovative piece of software.

Description

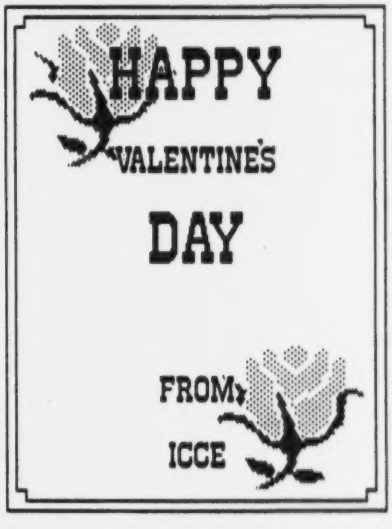
The Print Shop allows you to make wonderful signs, letterheads, stationery, greeting cards and banners. It is so simple to learn to use that you will be creating your first product within minutes.

The program is divided into separate sections for greeting cards, signs, letterheads, banners, backgrounds and graphic designs. Once you have chosen a section, you can first choose from 60 graphic designs such as animals, holiday icons, border prints and school-related designs—or create your own. Next you can select from eight different print fonts in two different sizes. With font names such as tech, block, party and typewriter, you get the idea of their versatility.

Before using the program, you need to call up the section titled "Setup," which asks you to specify your printer, printer interface card, printer slot number and number of disk drives. If you don't know these things, you will need to find out before you can print hard copy. It is important to note that this program does not work with all printers. It does work with the Epson RX80, MX80, MX100, FX80, FX100; Apple dot-matrix; Imagewriter; C. Itoh; NEC 8023A; Star Micronics 10x and 15x; Okidata 92 and 93; and Apple Scribe. If your printer is not listed here, check with your dealer or call Broderbund before purchasing the program.

Suggested Uses

This review can't begin to cover all the uses of this program. It can be used to enhance bulletin boards (no more long hours spent cutting out letters for slogans and titles), to advertise school activities, to make posters for upcoming events or student body campaigns. Students use it to generate personalized greeting cards for a fundraising activity. On a holiday such as Valentine's Day you will need to dedicate a printer for the cause. Banners can be made to advertise or celebrate any number of events. It is an eye-catching way to recognize contest winners, the fact that your school principal is having a birthday, or any other special event. In the face of tight school budgets, this disk can do some of the things the district's graphic artist used to do. Every instructional media center should have one. Actually, its uses are limited only by the imagination. Staff members will want to design their own personalized letterhead and then reduce it to note pad size. The cover pages of district-developed curriculum or district reports will soon show the effects of *The Print Shop*. Students, of course, will think of uses not yet conceived of.



Strengths

This program does exactly what it professes to do. It is easy to use and handles errors well. Menus guide you through each section and the prompts are so well written that it is actually fun to use. If you do make a mistake or want to change your design, it is easy to do. You can also stop the printer if you decide you want to make some changes after the print command has been engaged. It doesn't require use of a graphics card nor does it require the user to know anything about graphics. The back of the disk includes a demonstration that describes some features of *The Print Shop*. The program was written for a color monitor but works nicely with a monochrome monitor.

A nice touch is the fact that 20 sheets of bright-colored computer paper and eight matching envelopes are included in the package. There is also an order blank, so you can order extra colored stock and/or colored printer ribbon. If you don't choose to order colored printer ribbon, a felt marker works wonderfully to enhance your products.

In addition to creating a very usable piece of software, Broderbund has also come up with a reasonable licensing agreement. The disk has a 90-day limited warranty. After that, Broderbund will send a replacement disk for $7.50. You are also allowed to make one copy of the program using a built-in utility.

Weaknesses

The program has very few weaknesses—it does what it was designed to do very well. The fact that it does not underline was a minor irritation when I tried to use it to make certificates (a logical use of such a program).

Using the KoalaPad or joystick allowed for a great deal of creativity in the creation of graphic characters, but I was disappointed in the size restrictions when using the graphic editor section of the program.

Be aware that this will be a heavily used disk—extra printer ribbons should be ordered so that replacements are available as the old ribbon starts getting faint. Also, recognize that it takes time for the printer to generate a banner, especially if you use the option to create solid instead of outline printing.

Summary

The Print Shop is an exceedingly well-written, useful program. It was designed to make posters, banners, letterhead, stationery and greeting cards, but with a little creative thinking you will find an almost unlimited number of uses.

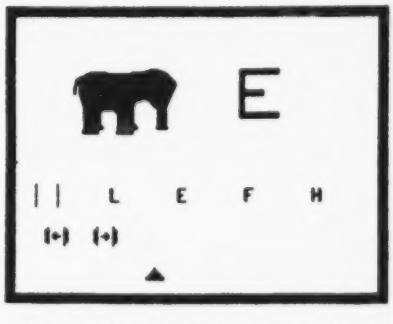
This program will probably be one of the most frequently used in your school. It is so easy to use, Broderbund advertises that use of the manual is unnecessary. They are nearly correct. It does not require any special equipment or knowledge to run through and print out your selection. This is a program I highly recommend. **END**

© Copyright 1985 ICCE

LETTERS AND FIRST WORDS
C & C Software, 5713 Kentford Circle, Wichita, KS 67220
Grade/Target Age Group: Ages four to eight years
Subject: Reading
Hardware Required: Apple II, II+ and IIe (48K RAM); Franklin Ace; single disk drive; color monitor recommended
Cost: $40
Reviewed by: Scott Turner, First Grade Teacher, Washington Elementary School, Vancouver, WA

Letters and First Words consists of three multi-level learning activities that provide practice in phonetic reading skills. A well-designed and versatile educational management program allows the teacher to control and monitor presentations to individual students.

The first activity, "A-B-C," presents a letter along with its associate picture. The pictures have been carefully chosen to represent the word's initial letter sound. The child uses the arrow key to position a box around the matching letter from among four choices, and then presses RETURN. If the correct choice is made, the picture then becomes animated for two or three seconds. If two incorrect answers are given, the matching letter is shown. Pressing the SPACE BAR will bring the next display. The highest level of "A-B-C" requires the child to match letters in different cases.



"A-B-C" presents a letter and asks children to match it to one of a group of upper or lower case letters.

The second activity, "Letter Sounds," presents a letter along with a number of example pictures beginning with that letter and sound. The child is then asked to choose another picture that begins the same way. Final consonant sounds and middle vowels are treated in higher levels of the game.

In the lowest level of the third activity, "Building Words," a picture is displayed and the child must choose the correct three-letter word. At upper levels of play, a picture is displayed and the student spells simple words.

It is the educational management program that sets this program above the rest. The teacher can make modifications to meet the needs of a variety of learning situations. In the least restrictive case, the program can be set up to require the user to pick one of the three activities. The presentation would begin at the lowest level of play and progress to higher levels as mastery is displayed.

The score-keeping features can be fully used if a student's name is entered before play begins. The computer will keep track of the score along with the current level of achievement, and save it on disk. Each time children use the program, they begin where they left off.

Current cumulative scores for an individual student or the whole class can be run off on a printer. Information on number of attempts, number correct on the first try, and number wrong is provided. One missing piece of information is the student's current level. This information is available in another section of the management program, but I found it cumbersome to have to call it up and write it on the printout.

Maximum program control can be gained by requiring that each name used match a name on your predetermined roster. Students can then have their presentation personalized. Separate entry and target levels can be set to meet the needs of individual students or to test for a specific skill.

Other modifications are available through the management program, including control over the selection of pictures to be used with each letter sound and the problem emphasis. The sound can also be controlled between full effects, minimal or none.

Letters and First Words works well with young children. Minimal teacher supervision is needed due to self-pacing, effective displays and ease of control. Directions and prompts can be given in words or through symbols that match the keyboard stickers.

In my first grade classroom it is up to students to set a mechanical timer that signals the end of their turn. This is a questionable method at best, given the user's age. Some type of timing option that would allow each student a given number of minutes to use the computer during each session would be a welcome addition.

Success in using this program is based on the assumption that children know what word each picture stands for, and that they have the correct phonetic pronunciation. The directions have an option to view the pictures used, but the display is not adequate for large group presentations. I wish that flash cards were provided. In many cases, the words for the pictures have to be taught to the children before they attempt the program.

While *Letters and First Sounds* cannot be considered a complete reading program, it can, when used in conjunction with other material, be a valuable resource to anyone who is teaching beginning reading. Its management options allow the teacher to control the presentation along with maintaining accurate records of student progress. This is the best phonics reading program I have seen.

© Copyright 1985 ICCE

DELTA DRAWING
Subject: Graphics programming
Producer: Spinnaker Software
215 First Street
Cambridge, MA 02142
Version: © 1982
Hardware: Apple II+ with 48K and one disk drive with DOS 3.3; printer (optional). Printer requires either a Centronics Parallel card or a Grappler card.
Target Age Group: 4-adult
Documentation: 80-page manual and eight "Fast Start" cards.
Back-up Disk: None provided.
Price: $59.95
Reviewer: David Lang, Eugene, Oregon.

To determine how heavily this software depends on external documentation, I began my review of this program by ignoring the explanatory booklet and simply turning on the computer. The screen menus easily led me to the beginning of the program and I found myself facing a triangular "delta" cursor with no idea of what to do next. Clearly I needed more information. Upon reading through the first six pages of the documentation, I learned that the Fast Start cards contained enough information to send me happily chasing turtles around circles.

It is this immediacy of DELTA DRAWING that appeals to me. In minutes I was creating programs that were visually exciting and often surprising.

DELTA DRAWING is a turtle-graphics-style programming environment. The user controls the cursor either in the graphics mode or in the text mode. In the graphics mode, the user gives directional commands to a turtle whose path is traced on the screen. In the text mode, the user enters the commands into a text program that the turtle then executes. Once a shape is created, it can be used as a foundation for other programs.

An example:

Command	Meaning	Function
D	draw	moves the cursor 1/4 inch
R	right	points the cursor right 30 degrees
1	name of first program	combines the above into a single command
press '1' 12 times	execute program #1 12 times	a 12-sided figure is created
2	name of second program	combines the 12 commands into a single command

DELTA DRAWING helps children learn about shapes, colors, compositions and patterns. It gives them opportunities to express their ideas and develop ingenuity through visual thinking in a friendly and forgiving atmosphere. By trial and error, children can explore an idea, and if they don't like the results, they can easily change it. The manual asserts that with DELTA DRAWING there are no mistakes, only unexpected results.

While older children should have little trouble learning to use DELTA DRAWING (most of the commands use) only one key), younger children will require a great deal of instructional supervision. The "no errors" learning environment encourages experimentation and learning in all age groups.

A ten-year-old friend who tested DELTA DRAWING enjoyed it and soon moved from drawing his ideas in graphics to writing the procedrues in the text mode. It was interesting to watch him think through the procedures for creating a triangle. After each try he would discuss what happened, what should have happened, and then he would try again. His enjoyment and learning were apparent even in the short time he was working with this package.

Weaknesses:

- DELTA DRAWING does not come with a backup disk and is copy-protected, so users cannot make their own.
- The command that turns on the printer should take at least two keys so that it cannot be accidentally activated.
- The manual needs an index.
- The erase command should be interruptable to prevent a disaster in case the "e" (erase) key is accidentally pressed.
- The text editor should be more powerful by allowing insertions and deletions anywhere within a program.

Summary: DELTA DRAWING is a good educational package. Its visual approach makes it very appealing. DELTA DRAWING achieves what it sets out to do: create an easy-to-use, error-free learning environment. It is fun, dynamic, visually exciting, and encourages experimentation. It is well packaged. The Fast Start Cards are sturdy and informative. The manual is reasonably well written, although the authors tend to alternate between addressing the adult helper and the child user.

While DELTA DRAWING is a powerful and stimulating package, it lacks the complexity and added functions of other Logo-type programs. I believe DELTA DRAWING merits close examination and comparison with other programs of its type on the market.

Publisher's Reply

DELTA DRAWING™ teaches a child procedural thinking, the concept of program nesting, and the relationship between a command and a graphic effect. Children never "make a mistake" with this program—they just produce unanticipated, but very pleasing, results!

Spinnaker aims to make learning fun and DELTA DRAWING does just that by guiding the first-time computer user into the world of personal computing.

Nancy Evans, Editor
Spinnaker Software

© Copyright 1985 ICCE

TURTLE TRACKS
(1982 software; 1983 handbook)
Scholastic, Inc., 730 Broadway,
New York, NY 10003
Grade/Target Age Group: 9 and up
Subject: Computer Literacy
Hardware Required: Apple II, II+ or IIe (48K). (See also note at end.)
Cost: $39.95
Reviewed by Marianne Handler,
South School, Glencoe, IL

Congratulations to the team of Tom Smith and Scholastic! *Turtle Tracks* has taken the best of Mr. Smith's earlier program and made it even better. It is well done, user friendly, and educationally sound. Mr. Smith had written in his earlier documentation that his program was meant to be a teaching tool and was designed expressly for that purpose. He feels that early computer experiences should parallel the introduction of other concepts that use manipulatives. *Turtle Tracks* therefore provides some instruction, some practice, a great deal of exploration and discovery time and an opportunity for children to realize that they are, indeed, in charge of the computer.

Like Logo, the commands are simple and as in BASIC, line numbers are required. The student can *run* (command:R) the program and see what s/he has created or he can *walk* (command:W) the program step by step to find his or her errors and debug it. Each chapter in the short but inclusive handbook is very specific in its description of the new commands, and includes a directed experience for trying them out. Then there are several opportunities for exploration following this exposure. *Turtle Tracks* makes use of a two-screen format. Programs are entered on the "blackboard." When the program is completed or the student wishes to see it at any given point, s/he need only enter R and press RETURN. The message RUNNING appears on the blackboard and then the Turtle Screen appears and the program is run. Pressing RETURN again will bring back the blackboard and programming may continue. In addition, the 80-page handbook lists all the characters available, some common errors and why they occur, as well as solutions to the variety of challenges in the book. Several pages are left for student notes. It would probably be better for students to keep their journals on other paper, however.

Students have an opportunity to work with both color and sound in this program. In a unique format, *Turtle Track*'s color chart combines the color of the turtle and a background strip in the same command. These can be combined with any of the 127 characters that the turtle can print. For example, if the program lines were to read

```
10 P#45;16      (PM;16 would also do it)
20 DF24         (DF stands for DRAW FORWARD)
```

the turtle will draw 24 M's in white on an orange background. If character #0 is used, a solid background can be drawn in the color of choice because character 0 acts as a blank.

Just as the P command changes characters and color, the command B (for Beep) brings sound and music. The numbers represent a note along a four-octave range and can also represent the duration of the note. Even rests can be built in by using the REPEAT LOOP command with nothing in the loop. It acts very much like a FOR-NEXT loop.

Turtle Tracks has two additional features. It is possible to enter a "Micromode" and use the "Microturtle" to create new characters. These new characters do, in fact, replace existing characters as long as they are in use and will remain until the computer is turned off and can be saved for use at another time. It was really neat to create my own shape and have that be the trail that the Turtle left behind. The second feature allows children to use the printer by using just two simple commands. LPAPER will print out the list of commands or instructions that have been given to the computer, and PAPER will print out the graphic that is currently on the turtle screen.

As with all programs, there are some drawbacks. The turtle can turn only 90°. TR (Turn Right) and TL (Turn Left) do not take any inputs. It is possible to draw a diagonal line but it requires a loop procedure.

In addition, *Turtle Tracks* comes with one disk. Scholastic's Customer Service department informed me that at this time it is not company policy to provide a back-up disk. The good news is that Scholastic will replace a defective disk for $5.00 even beyond the ten-month period (following the 60-day warranty period) that is described on the Warranty Registration Card.

In *Mindstorms*, Papert talks about the difference in making a mistake in content areas and making a mistake on the computer. Working with *Turtle Tracks* makes that point. It is not a terrible thing to make a mistake in a program. A program is rarely, if ever, correct the first time it is displayed; going back and debugging the program is a natural step.

Students reap benefits working with this kind of program. It will help them to look ahead and plan a project, find errors and correct them, and share their ideas with other students. Scholastic recommends this program for use with students nine and older. I disagree. Second and third graders can be introduced to the beginning commands, use them well and learn how to plan and debug the programs they are writing. The commands are simple enough for children to plan interesting graphics. Older students will begin to use the variables and find their use adds an exciting dimension to their programming. They can also create and save new commands for the turtle to use. The planning skills, problem-solving experiences and comfort in working with the computer that primary and intermediate students can experience in working with *Turtle Tracks* will be an excellent component of a total Computer Literacy Program. This turtle may not be as sophisticated as the Logo turtle, but *Turtle Tracks* will serve very well in an elementary school setting.

Note: *Turtle Tracks* is currently available for the Atari 400/800 (24K cassette, 32K disk); for the TI 99/4A (32K cassette); and for the VIC-20 (13K cassette). Disks are $39.95; cassettes $29.95. They are very similar to the Apple version; check with Scholastic for the exact differences. In addition, *Turtle Tracks* will soon be available for the Commodore 64 and the IBM-PC. Those prices may be somewhat different and are not available at this time.

© Copyright 1985 ICCE

MICROCOMPUTER PHYSICS LAB—HEAT
Cross Educational Software
P.O. Box 1536
Ruston, Louisiana 71270
Grades 10-12

Apple II+ or IIe
$60.00

On a slightly lower scale, in both price and execution, we have the programs from Cross Educational Software. Dr. Mark Cross has written articles about educational philosophy in *Call A.P.P.L.E.* and elsewhere and has a firm foundation in physics as well.

Rather than an all-inclusive program dealing with many measurements, Cross has taken the method of providing modules. The one I reviewed was dealing with heat. There is another one dealing with light which measures the time interval between flashes of light or something blocking the probe, such as a cart moving down an inclined plane or a pendulum. For $60.00, you get a diskette, a three-ring binder for documentation, and the necessary hardware for four temperature probes. To use them, however, you have to put them together, which is not really that hard. It involves soldering four wires onto a 16-pin plug and the other ends to two thermistors (which look like little blobs of metal with two wires coming out). Everything is provided except the solder and a soldering pencil.

Rather straightforward and simple, the programs on the disk allow you to calibrate the thermistors (each one will be slightly different when you get them hooked up), get a continuous readout of temperatures on the screen, or, most importantly, see the temperatures displayed both in digits and in a nicely done graph on the screen. The range of the temperature graph can be set as narrow or wide as you want, and the time interval between temperature readings is adjustable between one and 1000 seconds. That means a graph can display 270 seconds (4.5 minutes) or as long as 270,000 seconds (75 hours). You could, for example, set up a heated object in a box with some specific insulating material and come back the next day (or three days later) and see how the temperature had dropped. A second thermistor could be reading the room temperature's variation at the same time and a third just be a control in a box of insulation. (A fourth thermistor is provided if you want to hook it up.)

© Copyright 1985 ICCE

Your graph would show all three lines in a quite readable form on the screen. If you wish to have a hard copy of the numbers, just turn the printer on before you leave and the numbers will be listed there instead.

The program could be easily modified to write to a file on the disk, although timing might be changed slightly.

You can do that sort of thing because Cross Software has no "protections" on the disk. You can copy it or modify it as you see fit. This program can easily be used by a student without much preparation, but full usage of this package assumes there is a real teacher out there who is willing to experiment with new applications. The manual gives outlines of five experiments for use and also includes listings of all programs used on the disk, including the assembly language timing routine, for you to learn from and modify.

I had a problem with the program when I first received it. I wrote a letter to Dr. Cross and had a new, corrected version back in the mail within five days. A letter accompanied it that showed I was dealing with a real, live person who cared about how his software worked for teachers.

Again, although this software does only basically one thing, it does it well and at a reasonable price.

Also available from Cross Educational Software: 12-volume *Physics* sets ($260.00 total or $15-$35 each); and *Physics Lab-Light* and *Physics Lab-Sound*, both at $60.00.

PRECISION TIMER	$39.95
GRAPHICAL ANALYSIS	$24.95
RAY TRACER	$24.95

Vernier Software
2920 S.W. 89th Street
Portland, Oregon 97225
Grades 10-12

Apple II + or IIe

David L. Vernier has taken the "module" approach, similar to that of Dr. Cross. In this case, however, the equipment does not come with the software.

The disk called *Precision Timer* is basically a timing device to tell when one of the paddle buttons has been pushed or released. In the operation of this disk, the activation is by light, rather than by pushing a button, although it could be modified to do that.

The disk is designed to operate with "photogates" in a module from PASCO (part #ME-9204). Each of those gates costs $75.00 and you will use two of them. An interface box to the game port costs $100.00 (part #AI-6575). On the other hand, you can build the interface box with *three* photogates using parts bought at Radio Shack for $24.12. Mr. Vernier gives instructions on what parts to use and how to put them together. I know how to solder but am weak on semiconductor devices. Even so, I was able to put together the whole thing in about three hours. The worst part was going around to five Radio Shack stores to get the 18 parts, since each store seems to only have one of each item. Phototransistor FPT-100 usually consists of a card on a rack with "Sorry, out of stock" right behind it. Other than that, putting the hardware together is fairly straightforward.

Once you have everything together, the software does the rest. There are 12 different modes of measurement, depending on how many light sensors you wish to use. Basically, each works by cutting off light to a very small sensor. This begins the timing, and then turning off another or the same one again stops the timing. You can drop a ball past two sensors and get the speed, past three and get the acceleration, let a pendulum go past one sensor twice and get the period, perform collisions with carts or airtracks and get all the data you need. You can also use the keyboard, without any other equipment, to make this program a quite accurate stopwatch (+ or − 0.0002 seconds).

One nice feature of the disk is that all numerical displays can be shown on the screen in numbers four times as big as the normal digits on your monitor. Thus the data collection can be seen from the back of the room for demonstrations or larger group labs.

One caution about this equipment: the third light sensor uses paddle button #2, which is also the pin used in most "shift key modifications" for the Apple. That means that it will appear that the shift key is being held down. One way to do the shift key mod is from the #1 pin of the chip just in front (toward the keyboard) of the game paddle port, to the keyboard #2 pin using spring clips (Radio Shack Cat. #270-372). Installation takes about 15 seconds, but more importantly for this experiment, *disconnecting* it can be done in about 5 seconds.

Up to 100 data points can be collected, and then average and variance will be printed out. For many of the experiments, further mathematical treatment is possible, specific for the usual use. For example, the pendulum experiments will give the data of period, frequency, etc. that you would normally calculate.

The 42-page manual contains a user's guide, a teacher's guide, modifications to the equipment and programs that can be made, and a complete listing of all programs with a table of variables. The disk is completely unprotected so back-up copies and modifications are easily made.

Graphical Analysis can interface with the data from the above experiments, collecting stored data from the disk directly. Or, you can enter pairs of data for up to 32 points.

The disk is a very quick way to visualize data. Putting in the points is one option. Using the Plot option, you can label the axes, put in the units to be displayed, plot just points, points with connection,

© Copyright 1983 ICCE

Characteristics of a Scientist (Copyright 1983)
Cygnus Software, 8002 E. Culver, Mesa, AZ 85207
Target Age Group: Grades 6-12
Subject: Science
Cost: $39.00
Hardware Required: Apple II, II+ or IIe, disk drive (DOS 3.3), 48K RAM, Applesoft in ROM
Reviewed by Jack Webber, Kirkland, WA

Description

Not only does this program give the characteristics of a scientist, but it also gives a good definition of science.

The program begins by questioning students on their definition of science. The one provided by the program suggests it is a process or method of answering questions. To add a touch of class, it quotes a French mathematician!

The user is then baited with the following screen:

What is a scientist?
1. Wears a white lab jacket
2. Wears glasses
3. Male
4. Excellent student
5. Brilliant in all areas

Do you agree? (yes or no)

After the user either falls for or resists this generalization, the program goes on to define the four main characteristics of a scientist:

1. Curious
2. Observant
3. Skeptical
4. Open-minded

Each of the four parts has a definition of the word, an activity in which the user must assume that characteristic in order to succeed, and a final summary statement of how the characteristic relates to being a scientist.

The activities are the main focus of the program and are very well presented. The user must become curious enough to solve a puzzle for Farmer Crabtree of Wahoo, Nebraska, and then observant enough to remember if the head of a Lincoln penny faces left or right. The activity for the skeptical part catches the student trying to solve an unsolvable problem, while the open-minded activity concerns a beaver that seems to be ten feet tall!

A short quiz follows the lesson; it consists of 11 true/false and multiple choice questions which do a really nice job of summarizing the whole program. The score is kept and a positive comment ends the lesson.

Strengths

The program holds student interest with challenging, thought-provoking activities and occasional humor.

Use of color, sound and good use of inverse for highlighting enhance involvement; the program doesn't drag.

Escape options are provided throughout the program; at the beginning of the program the user may choose to run specific sections or skip others.

Many points in the program can be easily elaborated on by the teacher; in fact, one section even recommends asking the teacher to demonstrate facts that invoke skepticism (hot water freezing faster than cold water).

The written documentation is complete, including answers and an outline of the main points covered.

Weaknesses

A few cutting remarks about teachers and programmers struck me as more distracting than funny.

A quote and reference from the Bible (I Thessalonians 5:21) could conceivably be offensive to some. There are a few punctuation errors—something that's always annoying.

Summary

For the cost, this program should be in your collection if you teach science. It can be used either with a group or with individual students. The humor is good, the pace enjoyable, the activities stimulating and the approach refreshing. It fills a need beautifully.

Recommendation

If part of your objective in science is to define the subject and give an idea of what a scientist is, then this is a good piece of supplemental software. It contains four secondary programs that could be used as springboards for a number of other related topics such as scientific method or optical illusion. For the cost, it is an inexpensive way to enrich your curriculum.

© Copyright 1985 ICCE

or "point protectors" of boxes around each point. A linear regression analysis can plot the closest fit straight line also. Other menu options give you the choice of exchanging the axes, changing any pair of points, and a numerical analysis section giving slope and intercepts.

The program doesn't do anything really fancy, but for $24.95 you get a very quick way to display lab data in a graphical form. The scales on the side can be done in "auto-scale" mode for quickness, or you can determine the value of each unit for clarity. Although this is not as instructive as learning how to make a neat graph with ruler and pencil, perhaps your interest is in teaching how to *set up* graphical displays and what they *mean*, rather than the mechanics of pencil and paper neatness.

Ray Tracer is an electronic blackboard. There is nothing on this disk that a good physics teacher couldn't do with a blackboard, chalk, straight-edge, protractor, compass, fantastic skill in drawing and three hours for each class period.

The key differences are in accuracy and time. I can draw a good picture of what a light ray will do at the interface between air and glass. But I can't do it accurately for both red and blue light, for any pair of materials, and for 15 different angles in under two minutes. I can *talk* about the "critical angle" where light bounces off of water, but it's more instructive to "sneak up" on it by changing the angle of incidence slowly until the reflection takes place on the screen.

Besides refractive indices between two media, other options from the menu of 15 situations deal with thin and thick lenses, compound lenses, plane and spherical mirrors. In each case, one white ray will be drawn, or you have the option of doing two rays (red and blue) to show the difference wave length makes. The programs can be run by students as a "game" since a source is picked arbitrarily and then a target is placed somewhere. The object is to hit the target, no matter what media you have to go through or reflect from or refract through. This sort of "gaming" can teach a lot of intuitive feelings for what optical devices do without a lot of difficult lab setups.

The 29-page manual has a user's guide, a teacher's guide, what optical laws are being demonstrated and a list of experiments and references. A complete listing of the programs and variables used are also included on this unprotected, LISTable disk. You are given hints on modification and are encouraged to do so, to make the programs fit your exact requirements.

If the drawings aren't fast enough for you, there is also a *compiled* version on the disk that runs 2-3 times faster, although modification or even listing is then impossible.

Again, at $24.95, this disk doesn't allow you to do anything you couldn't do before, except improve accuracy, save time, and make optics more interesting. Perhaps such things are attractive to you.

Also available from Vernier, but not reviewed: *Orbit, Projectiles, Kinematics, Charged Particles, Vector Addition,* and *Wave Addition,* all at $24.95. (These programs have also undergone recent revision.)

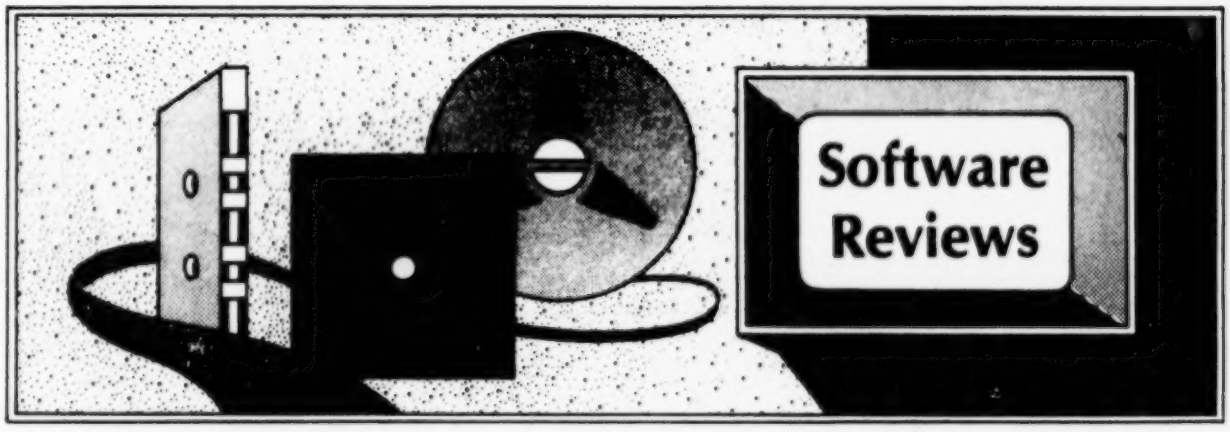
Author's Reply:

Graphical Analysis has undergone major revisions with several new features added for the same price.

- 200 pairs of data may now be entered, instead of 32.
- Data is rounded before it is displayed—the number of significant figures is controlled by the operator.
- The appearance of the final graph has been improved.
- Semi-log and log-log graphs may be displayed.
- A hard copy of the graph can be made on any standard 80-column printer.

END■

© Copyright 1985 ICCE



GEOGRAPHY SEARCH

Reviewed by Mollie L. Cohen

(See also related article by Mollie Cohen on page 11.)

Producer: McGraw-Hill/Computer Learning Connection. © 1982.

Cost: $180 (diskette, 25 workbooks, and Teacher's Manual).

Subject: Social Studies, Geography.

Grade Level: 5-12.

Medium of Transfer: 5¼" flex. disk.

Required Hardware: TRS-80 Model III (32K); or Apple II (48K) 3.3 DOS

Instructional Techniques: Simulation, group decision-making.

Your students are crew members of an old sailing ship, seeking the fabled City of Gold on an unknown continent across an uncharted ocean. The wind is blowing from 195° at 32 miles per hour. A student sailor carefully records this data from the ship's instruments into the Captain's Log. At noon, as the ship's clock reads 12:15, a second crew member records the ship's longitude location: "15 minutes west of Vesuvia." As a third student sailor checks the North Star to read the ship's latitude, a fourth takes a sounding of the ocean depth while another checks the food supply. Armed with their individually collected data, the student sailors meet and share information about their ship's position and together decide on the next day's plan of action.

What I've just described is not a real ocean voyage, of course, but a simulation of one called **Geography Search.** The program is one of five in the "Search Series" published by the Webster Division of McGraw-Hill Book Co. The series was developed by Thomas F.F. Snyder, a science and math teacher from Cambridge, Massachusetts, who has also formed his own company called the Computer Learning Connection. The series is being promoted for use with students in grades 5-12, but seems most suitable for average-ability classes in 6th through 8th grades (or for lower-ability students in grades 9-12). *Geography Search* would be especially useful in 6th or 7th grade geography classes and is also suitable for 8th grade U.S. History.

Program Objectives and Format

Each Search program has three main objectives: (1) to teach concepts related to subject content such as geography; (2) to teach information-processing skills such as reading, recording and interpreting data; and (3) to improve student group-process and problem-solving skills. The components of each program include the Searchbook, or student workbook which provides story and content information; a diskette for the microcomputer which provides and manipulates data for the simulation; and a Teacher's Manual (which comes in a 3-ring binder with the diskette and a sample Searchbook).

The Teacher's Manual recommends that students first read the Searchbook all the way through to become familiar with the story line and geography concepts as well as with the instructions for the simulation. Student groups of 5 or 6 members each are then selected as the crews of up to 6 sailing ships. The groups take turns at the microcomputer, getting data about their ship's present course and entering information based on crew decisions. Each turn at the microcomputer simulates one day of the voyage and is followed by meetings where crew members share information and reach decisions. The process is repeated for as many "days" as necessary until each group gets to the City of Gold and back.

The Searchbook and Program Content

Before assessing the accuracy of subject content, one has to suspend reality somewhat and accept the interplay of fact and fiction which is a deliberate part of the *Geography Search* simulation. Even the fiction, however, suggests reality: for example, students set sail from an island-nation called Vesuvia which vaguely represents Europe; they reach a New World continent which suggests the Americas, although much smaller in size. Although historical time is not an integral part of the program, the Teacher's Manual identifies the simulated voyage as that of an "ancient" sailing ship. The voyage, however, would be more logically placed historically in the high middle ages or early modern times —a relevant fact to history teachers who might use the program.

© Copyright 1985 ICCE

Since the program is, after all, a geography search, the geography concepts used in the simulation *are* closer to fact. For example, the circular winds which blow counterclockwise and affect student sailing decisions resemble real Atlantic tradewinds. The simulation makes use of other real geography facts, too, such as the relationship of time to longitude. The Searchbook explains these and other geography concepts in clear, simple language which most students can easily understand. This is an important strength of the program. Teachers may still want to supplement the material to make it more complete—by relating the discussion of wind patterns, for example, to the use of terms like "westerlies" or "easterlies" to describe them. The use of "minutes" for longitude location in the simulation provides an opportunity to explain the system of subdividing latitude and longitude degrees—which is not in itself covered in the program.

The Diskette and Skill Development

Although it took a while to figure out how to enter today's date when I first ran the program, there was smooth sailing after that. The diskette is programmed so that as members of each crew take their turn at the microcomputer, they are given a choice of actions—whether to go ashore, follow the coast or sail away. If they choose to set sail, they must indicate the direction and duration. At the end of a turn, the microcomputer flashes new data very quickly on the screen, about the wind direction, North Star position, ocean depth, and so on, which will be needed to make a decision about action for the next turn.

The process seems confusing at first, but after a few turns the pattern becomes familiar; it then becomes easier to know what data to look for and how to record the data on the worksheets. It also becomes easier to use the recorded data in making decisions: for example, students gradually realize that if wind is blowing from the North, they must decide to sail in another direction or, if they can't see the North Star, they must be too far south.

The quality of the skill practice derived from this process is excellent. The simulation provides far more in-depth understanding of direction and location concepts and skills than would be possible with more traditional non-interactive skill materials. Latitude and longitude are no longer just flat lines on a map or globe but positions related to observable changes in the solar system which are "experienced" through the simulation.

In addition, the simulation process fosters development of information-processing skills on several levels. Here students go beyond the basic reading, writing and vocabulary-building activities of the workbook to locating and recording data (research skills) from the microcomputer. As they prepare for the next turn, students analyze, synthesize and evaluate this data, identify alternatives and anticipate consequences, thus using higher-level critical thinking and decision-making skills.

Interpersonal Skills Development

When sailing information is flashed on the screen at the end of each crew's turn, too much data is displayed much too quickly for any one student to read and record it all. Thus, a need is created for student cooperation and a division of labor. It is apparently the intention of the developers that this dilemma be resolved by students themselves who will—either spontaneously or through trial and error—parcel out the various roles of wind recorder, depth recorder, etc. among themselves. The teacher is cautioned in the Teacher's Manual not to get involved in this process or to answer student questions but rather encourage students to find their own solutions. The objective here of encouraging self-sufficient teamwork and problem solving through group interaction is to be applauded. As a teacher, however, I would have preferred that the Teacher's Manual provides some guidance on how to get the process started, or at least on how to evaluate whether the group interactions are working as well as possible.

Teaching with *Geography Search*

The Teacher's Manual of *Geography Search* includes a Record-Keeping Chart for recording each crew's progress after each turn. Also included are discussions of the overall procedure and learning strategy, a good illustrated sample of a typical turn at the computer, as well as lists of the geography concepts, vocabulary and questions (without answers) covered in the Searchbook. The list of supplementary activities provided would be even more useful if it were organized by level of difficulty.

Because the use of microcomputer simulations will be a new experience for many teachers, however, some practical suggestions on how to use them in the regular curriculum would probably be welcomed. Here this program does fall short. The Teacher's Manual of *Geography Search* does not include, for example, suggestions on how to introduce the simulation in class or when to plug it into the social studies curriculum. There are no concrete suggestions, moreover, for follow-up activities by which students can be helped to see connections between the simulation experience and course content.

The Manual does suggest that after students read the entire Searchbook through once, two sessions be allotted per week—one for rereading a section of the workbook and the other for using the disk and microcomputer. Although this may be possible in some classes, it may not be realistically possible for many social studies teachers to devote two out of five class periods per week for the simulation over an extended length of time. It is also not clear how many class periods or weeks the simulation should be expected to take. Precise estimates would not be possible since the decisions made by students affect the success of their voyages and thus the time needed to complete them. Nevertheless, some ballpark estimates of time needed would be useful to teachers who must fit the simulation into a larger curriculum plan.

© Copyright 1983 ICCE

Recommendations

Geography Search would be especially useful with an introductory unit of 6th or 7th grade world geography, where sections of the Searchbook could be assigned alternatively with related material in a standard geography text. The actual simulation could follow once or twice a week, with review of the Searchbook when needed. *Geography Search* could also be effective in U.S. history classes with a unit on 16th-century European explorations; such a simulation could give students some appreciation for the experience of exploring an unknown world.

Despite some minor flaws, *Geography Search* is an excellent program for social studies and is definitely recommended. The simulation is particularly good for providing "active" learning and practice of academic skills in the context of geography or history subject content. The program also provides a vehicle for developing student interpersonal skills—an important goal of the social studies, and one which is not possible with many other existing social studies materials.

All five programs in the *Search* series will use a similar format to achieve similar objectives with social studies content. The four other programs—*Community Search, Energy Search, Geology Search* and *Archeology Search* —are expected to be available in the fall of 1982.

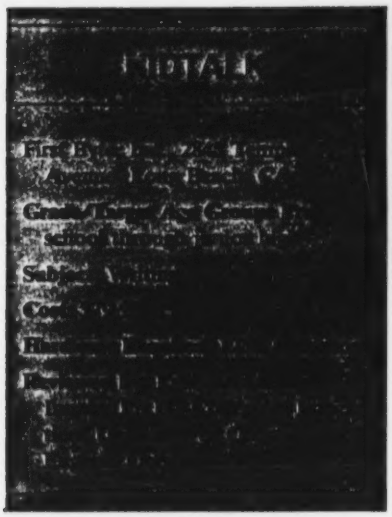
[Ed. note: All five programs in the Search *series will use a similar format to achieve similar objectives with social studies content. The other four programs—*Community Search, Energy Search, Geology Search *and* Archeology Search*—are now available for Apple II+ 48K, 3.3 DOS and TRS-80 Model III, 32K at $180 per* Search.*]*

© 1982 Mollie Cohen. May not be reproduced without permission of the author. **END■**

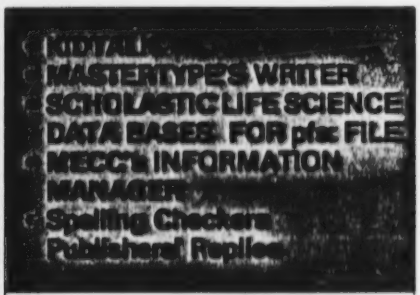
© Copyright 1985 ICCE

Software REVIEWS

Linda Rathje, Editor



If you add synthesized speech to a program for word processing, what do you get? *KidTalk* is one of the first answers to this question. Designed for the Macintosh computer, *KidTalk* incorporates the speech technology published earlier by First Byte as *Smoothtalker*, but in *KidTalk*, the synthesized speech of *Smoothtalker*'s male and female voices are packaged as a talking word processor for children.



A Writing Tool

As a word processor, *KidTalk* has all the basic functions needed to manipulate text, and it employs the editing commands familiar to Macintosh users, e.g., cut, copy and paste. Students write in a typical Macintosh "writing box" and are able to access any of seven pull-down menus, each accompanied by a picture.

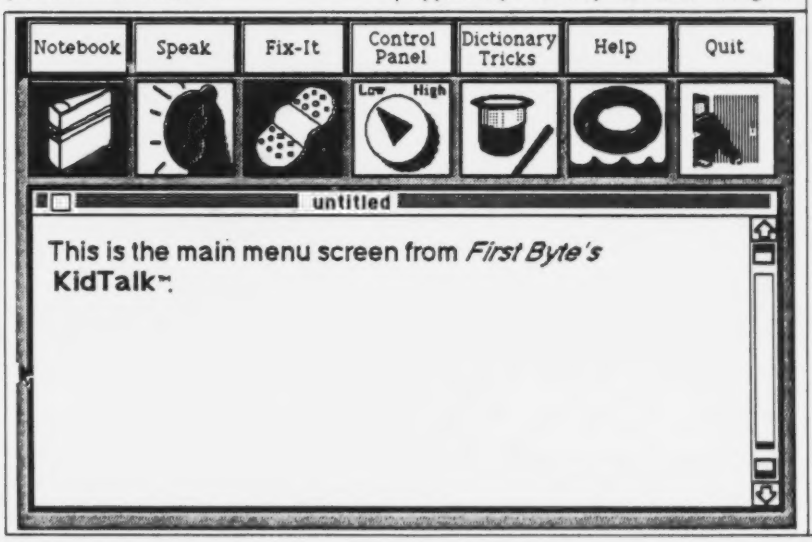
KidTalk is clearly easy to use. It is not, however, very flexible. Writers have the choice of three type sizes but no options as to the appearance of the font. Although this may seem to be a fairly trivial criticism, youngsters reared with a Macintosh have come to expect such features. Of greater importance is the program's lack of printer options. There is apparently no way to alter margins (other than using the space bar while writing) or even to underline. It can certainly be argued that such features are not necessary for young children. *KidTalk*, however, is billed as a program appropriate for students through junior high school and many of the suggested applications (e.g., stories, reports, plays, poems) could benefit from greater flexibility when it comes to printing.

A Talking Tool

But *KidTalk* is not merely another word processing program. It's a writing tool that talks. And it does so without necessitating the purchase of any additional hardware. This is something many of us have been waiting for.

KidTalk uses speech in two ways. First, the user is provided with verbal instructions on how to use the program, as well as ongoing assistance when needed. The program has a fairly extensive guided tour, in which its key features are explained by a parrot and demonstrated on screen. The guided tour is extremely well done, both visually and auditorially. My young users of the program were able to grasp the basics from this tour, although it must be admitted that neither were novices on a Macintosh. An improvement would be to make the guided tour interactive. The youngsters were constantly reaching for the mouse, wanting to try what had been demonstrated without waiting for the tour to end.

Voice is also used to provide ongoing assistance. Writers who are unsure of how to accomplish some editing task, for example, can access the "SOS" picture menu for help. Explanations, with accompanying demonstrations, are both simple and clearly presented. This is one



of *KidTalk*'s best features, making it possible for even young users to become fairly independent on the word processor without the need for adult intervention when a question arises. In checking actual utility of this assistance I asked a seven-year-old novice user to "figure out" how to get rid of a specific sentence in her story (without using the delete key to remove it letter by letter). She quickly located the SOS menu and accessed the verbal directions on how to "cut." The directions were sufficient to teach the function, but she had to go back to listen a second time before accomplishing the necessary sequence of steps. Her response to this assistance? "I think most kids would find it easier to ask their parents." So much for independence!

The second way *KidTalk* uses its built-in speech technology is really the most interesting. *KidTalk* has the ability to say anything the student writes. And for the vast majority of words written, what it says is correct. Child writers enjoy this feature immensely, feeling the added dimension of voice somehow makes their words really come alive. This feeling is enhanced through the many options available to control the synthesized sounds. The user can choose between a male or female voice and alter the volume, pitch, tone and speed. For words that *KidTalk* does not speak correctly, the writer is encouraged to enter an altered spelling into the "sounds like" dictionary. For example, rewriting the word library as "lie brary" will get rid of the unwanted short "i" sound. Words derived from a foreign language are also problematic, so the manual warns that a word such as Chihuahua should be entered into the dictionary as "Chi wah wah."

Educational Applications

Of possible value for a wide range of students is the use of *KidTalk* as an aid to proofreading. Writers have the option of hearing a sentence immediately after it is written, or they can choose to hear a whole section of text at one time. Either way the auditory feedback can serve to highlight typographical errors and misspelled words, reveal grammatical inconsistencies, and prompt recognition of redundant word usage. Two of my test users found the speech function of *KidTalk* to be quite helpful in revising their compositions. The seven-year-old found she had spelled the word "might" as "migt" and the five-year-old recognized that she had started each sentence with "and."

Another interesting application for *KidTalk* is the development of talking stories or plays. Not only can writers control the characteristics of the voice (sex, pitch, tone, etc.) but they can also change these characteristics as frequently as desired while writing. This means, for example, that one voice can be used as a narrator and other voices for dialogue. In a similar fashion, different voices can be created for the various characters in a play. The writer's choices are saved along with the text, so that future "readings" of the story or play will incorporate the choices made.

A final application is, of course, as an aid to improving reading fluency. Paragraphs or stories can be typed into a file and then presented individually to students for reading practice. When unsure of a word or phrase, the student need only highlight it with the mouse, then have the computer say it by accessing the "Speak" menu. In this manner *KidTalk* might be easily integrated with the language experience approach to reading instruction as well as more skill-based or remedial approaches.

Limitations

For any of the above applications to be practical, accuracy in speech production is critical. In general, the voice quality in *KidTalk* is intelligible and relatively few words are spoken inaccurately. That there are any inaccuracies at all, however, places some limitations on the program's use. Its utility as an aid to proofreading, for example, is somewhat hindered by the possibility that correctly spelled words might be targeted by the student as incorrect (and thus needing to be altered) upon hearing them mispronounced. The extent to which the program might actually encourage and reinforce certain types of spelling errors needs to be considered when structuring how the program is used.

Users of *KidTalk* will probably debate whether or not the quality of the synthesized speech is adequate for instructional purposes. The reality of the situation is that you can understand the computer fairly well if you know what it is saying! Being able to read along with the voice helps tremendously. Even adults I shared the program with often could not make out what was being said if they were not looking at the screen. While listening to one of the stories provided on the disk I asked my five-year-old if she understood the words. Her reply was hopeful: "I got one of the words—beach."

Much can be done to improve the overall intelligibility if you type extra spaces between the sentences. Without these spaces the words and phrases seem to all blend into one continuous stream. Unfortunately, teaching students to type four spaces between sentences would result in negative transfer to other word processing programs, not to mention a printed copy with more blank holes than is aesthetically pleasing. This is probably trivial to fix programmatically, and I would recommend the publisher consider adding more sound spaces between sentences.

Computers in Education

Master's Degree and CAGS programs designed to prepare professionals to integrate computer technology effectively in educational settings.

Three areas of specialization:

- ☐ **computer coordinator**
- ☐ **computers and special needs**
- ☐ **computers in the elementary and middle school classroom**

Courses are available throughout the year.

The computer coordinator specialization may be completed in a summers-only format.

Qualified non-degree students may take individual courses.

For information contact: Angeline Ferris, Lesley College Graduate School, Division of Education and Special Education, 29 Everett St., Cambridge, MA 02238

(617) 868-9600 ext. 362

LESLEY COLLEGE GRADUATE SCHOOL

Advertisement

My last major concern has to do with the recommended age range. The program authors suggest an age range of preschool to junior high for *KidTalk*. I would estimate its greatest value is in the elementary school grades. Although preschool kids might find it fun for practice on letter names and sight words, most would probably not be independent in their use of the program. *KidTalk* would have greater value for this population if it contained an option for sounding words out phonetically, rather than spelling them letter by letter. The older children I shared the program with thought it was "cute, but not very practical."

A few other concerns will be mentioned briefly. First, it is sometimes irritating to have to wait for the computer to finish speaking. This was especially notable when accessing the help menus. Once children had the answers to their questions they wanted to return to work. The mouse and keyboard are inactive, however, until the computer has finished the predetermined discourse.

Second, there is apparently no way to eliminate the voice. This could cause problems in the classroom unless earphones were available for the Macintosh. And third, the handbook, though generally well written, uses only male pronouns to refer to children. In a period of time when the issue of sex equity with respect to computer usage is widely discussed, it would seem appropriate to eliminate this vestige of sexist terminology.

Conclusion

KidTalk is a very laudable effort. First Byte should be commended for integrating the Macintosh's capability for speech with a program that is well designed, easy to use and educationally relevant. For the computer to reach its potential as a learning tool, we need software designers willing to push the limits and explore the unknown. This is not easy, and products that go beyond the commonplace may have failings that limit their potential utility. That this is true with *KidTalk* does not diminish its importance as a significant milestone in educational software.

Publisher's Reply

When First Byte developed the SmoothTalker technology, it also committed considerable resources to produce pioneering and educationally exciting applications for children. We believe a program such as *KidTalk* is a look into the future, when computers cannot only talk, but listen.

Currently SmoothTalker is the highest quality, affordable speech synthesis technology on the market that does not require a board. It doesn't sound like your next-door neighbor yet (that doesn't seem to bother kids) but First Byte is committed to improving the speech quality and has a consumer policy for all products.

KidTalk is the only educational speech product available that allows children the optimum control of being able to switch back and forth between a male and female voice within one document and manipulate the voice control of volume, pitch, tone and speed. It may not be the type of word processor an older child would choose to use to format a term paper. It was intended to be a simple word processing program for creative writing. However, if the user is an experienced Macintosh pilot, s/he can boot with another program to access the finder level, move the file to a more sophisticated word processor, format the document and then move it back to *KidTalk* for proofreading as ordinary text.

Reviews

Linda Rathje, Editor

Story Tree
Graphics Exhibitor
South Dakota
Dancing Bear

Teacher, Kids and Logo

A comparative review of 3 computer dictionaries

STORY TREE

Author: George Brackett
Scholastic, Inc., 906 Sylvan Avenue, P.O. Box 2010, Englewood Cliffs, NJ 07632
Target Age Group: Ages nine and up
Subject: Writing
Hardware Required: Apple IIc, Apple IIe, Apple II Plus (48K), or Apple II (with 64K or Applesoft in ROM); IBM PC, IBM PCjr (128K); Commodore 64.
One disk drive, monitor or TV, printer optional.
Cost: Retail edition, $39.95, includes Master Disk, Story Disk and User's Handbook.
Teacher's edition, $59.95, includes 2 Master Disks, 2 Story Disks, User's Handbook and Teacher's Guide.
Reviewed by: Lynne Anderson-Inman, Ph.D., College of Education, University of Oregon, Eugene, OR

Everyone loves a good adventure story, especially when the reader plays a role in deciding how the adventure will unfold. George Brackett's *Story Tree* capitalizes on this interest and takes it one step further. What could be better than ***reading*** "choose-your-own-adventure" stories? Well, ***writing*** them, of course! *Story Tree* lets you and your students do just that.

Story Tree is an exceptionally well designed piece of software that facilitates both the writing and reading of "interactive" stories, i.e., stories in which the reader makes choices determining the direction of events. Although an obvious application for a writing tool of this nature is in the creation of adventure stories, *Story Tree* lends itself to the development of other interactive genres as well. Provided as samples on the Story Disk are a somewhat silly jungle adventure in search of the magical Marigold Mine, a well researched and interesting article on the creature known as Bigfoot, and a mini data base of extremely brief book reviews organized by topic. Each uses the program's capabilities in a slightly different way and together they help to illustrate some of its applications.

The *Story Tree* program takes its name from the branching capabilities of the stories that can be created with it. When writing with *Story Tree*, the writer has three options for linking pages together:

a) A story can CONTINUE from one page to the next, as it would in any story;

b) The reader can be given a CHOICE from among two to four options, with each option leading to a different series of events or type of information; and

c) The story can be made to branch by CHANCE, leading in one direction some of the time and in another for the remainder.

If used creatively and in combination, the resulting stories can be read and reread without being repeated in exactly the same sequence.

Designed for children as young as nine, *Story Tree* is a program that has been constructed with care. The main menu provides the user with four choices: (a) Read a story, (b) Work on a story, (c) Print a story and (d) Do disk work. Each choice leads to an appropriate submenu or a listing of the available stories, and ESCAPE always brings you back to the main menu. When reading a story, the user can elect to read it from the beginning or s/he can choose another place to start by browsing through the page menu. The latter is especially useful when rereading stories, as it allows curious readers immediate access to different events and alternate endings. It is also useful when editing stories, as the page-by-page access to different parts of a story helps the writer to locate pages needing clarification or story lines needing development.

When writing a story, the user is prompted through three operations for each page: (a) writing a page title (to allow access through the page menu); (b) writing text for the story; and (c) indicating how the page should be linked to successive pages, i.e., will it CONTINUE to another page automatically, branch by CHOICE or branch by CHANCE? After making a selection on the latter, the user is asked a sequence of questions which enables the program to create the appropriate connections between and among pages. For example, if branching by CHOICE, what choices are provided to the reader and to which page should each choice lead? Although the above may sound complicated, doing it is actually fairly simple, in large part because of the excellent prompts presented to the user on screen. For writers who haven't thought out all the details of a story (e.g., only one of three alternative choices has actually been conceptualized), temporary choice descriptions and page designations (letters or numbers, for example) can be used and then changed to something more descriptive when details have crystallized. As they are being created, all *Story Tree* stories are saved on a story disk, allowing for later reading or editing. Stories can be altered or expanded at any time, thus encouraging authors toward greater detail and increasingly complicated plots.

Story Tree stories can also be printed at any time. Two *Story Tree* pages are printed on an 8½ x 11 piece of paper, and each section is marked with a page number. When cut and appropriately arranged, small books can be made from the printed pages. The capacity to print out stories is useful to students in at least two ways. First, it allows stories to be shared with family and friends. Second, having a printed copy enables the stories to be read and reread when the student is not near a computer, thus extending the educational value of the program and promoting the transfer of learned skills to noncomputerized settings. The reinforcing nature of the printed stories would be enhanced, however, if an optional large print mode were available. It

© Copyright 1985 ICCE

is important for young students, as well as for older students with reading and writing problems, to see a big payoff for such a big investment of effort.

Suggestions for Use

In preparing this review I used *Story Tree* with two students (ages nine and 14), and I talked with teachers who had used the program with small groups of students or entire classes. The following suggestions for using *Story Tree* represent a blending of ideas from these experiences and discussions.

1. Prior to introducing *Story Tree,* it seems best to ensure that students have had experience with "choose-your-own-adventure" stories.* Teachers found that students who had had previous exposure to this type of story in book form approached *Story Tree* enthusiastically. Those who had not had this exposure were often unclear as to why a story should be read more than once, and were more confused about the structure of *Story Tree* stories, especially when writing them. If grouping students for work on *Story Tree,* take into consideration their experience with this type of writing. Students with little or no experience with "choose-your-own-adventure" stories should be grouped together, otherwise they tend to be overwhelmed by their more enthusiastic and confident peers and usually end up as passive participants in the creative process.
2. Collect, swap and exchange *Story Tree* stories created by students. One of the best ways to be introduced to *Story Tree* is for students to read sample stories, either in pairs or as a small group. Although sample stories are provided on the disk, they are not appropriate for all levels of students and are not written by children. The 14-year-old I shared the program with felt the stories were too juvenile, and went on to create a complex James Bond-type spy story. The nine-year old felt Magic Marigold Mine was funny but complicated, and she couldn't identify with the report on Bigfoot at all. One of the best features of *Story Tree* is that it will support a wide variety of story types on an unlimited number of topics. Developing a library of stories for students to explore prior to writing their own will help to show off the program's possibilities and broaden its use.
3. Before writing *Story Tree* stories, it is especially helpful to teach students to map out or flowchart the sequence of events or information. This can be introduced using experiences from students' lives (to help convey the idea of choices and the impact of choices made on future events) and then by translating these experiences into the terminology used in the *Story Tree* program. The manual provides an excellent tutorial on how to write *Story Tree* stories and uses a mapping technique to illustrate the different types of page connections. Most teachers found this helpful in teaching students how to plan and keep track of their stories. (See figure 1 for the map used in this tutorial.) In general, those students who took the time to map out their stories were able to finish them; those who did not often ended up in confusion.
4. Integrate the use of *Story Tree* with practice on word-processing skills. Built into the program is an excellent and easy-to-use word processor. It can be used to introduce students to the principles of word processing or as an alternate arena in which to apply already learned word-processing skills. Although most of the word-processing commands in *Story Tree* are fairly standard, teachers will need to pay special attention to those that differ from word-processing programs students are already using.
5. If using *Story Tree* in a classroom setting, provide students with a five-minute warning before the end of the period. There is no way a

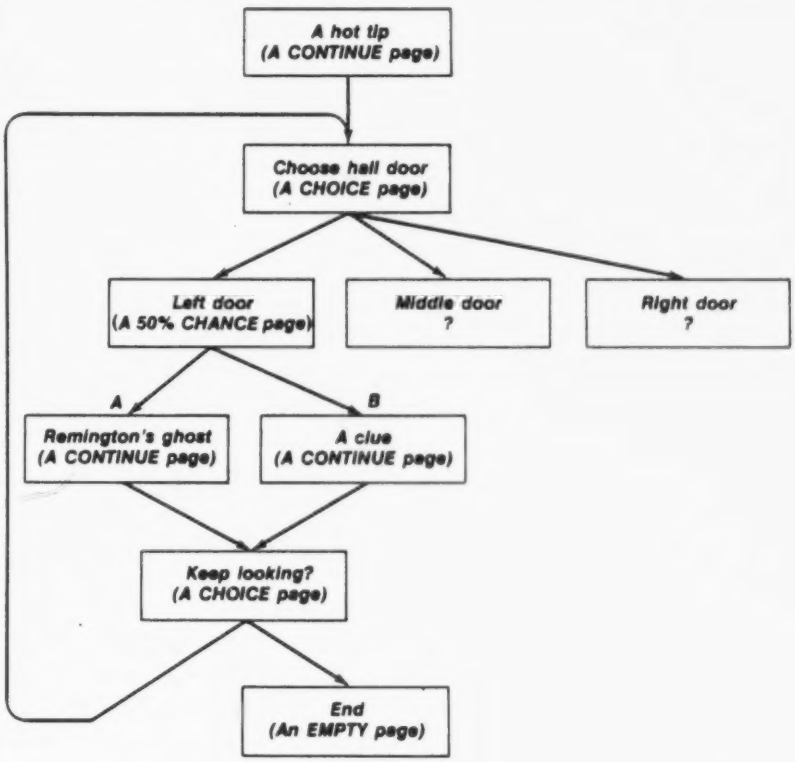


Figure 1.
mapping the sequence of events when writing Story Tree stories.

*Several book publishers produce books of this nature with varying levels of reading difficulty: *Choose-Your-Own-Adventure* by Bantam Books, *Twistaplot* by Scholastic Books, and *Which Way?* by Archway Books are three series that can usually be found in bookstores and in school or public libraries.

© Copyright 1985 ICCE

page that is being worked on can be saved before completion. This means that in order to save any text that has been written, the student must also designate the type of page connection and the destinations for each option before being able to stop. This requires some time—hence the need for a five-minute (or longer) warning.

Other Applications

Story Tree is not limited to the language arts or English classroom. And even in these classrooms, the program is not limited to the development of stories. The Teacher's Manual which accompanies the educational version of *Story Tree* does an excellent job in suggesting other uses for the program and even includes guidelines for creating special types of "shells"—frameworks of pages linked together for specific purposes. A sampling of these other applications is enough to highlight the underlying potential of *Story Tree.*

In the area of social studies, for example, groups of students could use the program to write an imaginary tour of another country or another era in time. Information could be organized around specific topics, as well as by location or date, allowing readers a different experience each time they took the "tour." In science, students might write a description of the circulatory system as seen from the "inside." Different branches of the descriptive report would address different branches of the system. A class poem could be generated using the random poetry shell, a framework that takes advantage of the branch by chance option. Each student writes a rhyming couplet on the same theme. When entered into the shell, the poem can be read and reread many different times with many different results. As a final example, the program can also be used to generate on-screen multiple-choice learning activities. Choosing the correct option leads to a "right answer" page and then on to another question. All incorrect choices lead to explanatory pages with individualized corrective feedback and then a return to the original question. It is suggested that students might develop these learning activities as well as play them. Although I was unable to find teachers who had implemented any of these ideas, or others from the manual, the ideas appear to be instructionally sound.

Making a Good Program Even Better

I've yet to see a software program that couldn't be improved upon in some way, and *Story Tree* is no exception. What is already a very good program can be made even better. The following suggestions for improvement come from teachers and students who have used the program extensively and therefore build on an in-depth appreciation for everything *Story Tree* already provides.

1. It would be helpful if *Story Tree* would automatically map or flowchart a story as it is being created. Younger writers, and those with special learning problems, often become "lost" in their stories, not being able to sort out which pages went with which story lines. Being able to consult a map of the story in progress would help to minimize this confusion and concentrate student energies on the writing process.

2. Students should have the option of seeing page titles on the printed version of their stories. As currently printed, page numbers are substituted for page titles. Since the pages are identified by title and not by number on screen, this substitution makes it extremely difficult to cross reference the printed version with the on-screen version. When editing, students often use the printed version to keep track of undeveloped story lines or to identify pages needing modification. Inclusion of page titles on printed stories would facilitate this aspect of the editing process.

3. It would be educationally helpful if *Story Tree* stories would allow use of the commonly available spelling checker programs. Teachers have become increasingly convinced of their value in promoting good spelling habits, and many teachers now encourage the use of such programs on all student writing. There is no logical reason why *Story Tree* stories should be an exception.

4. And finally, students should be directed to develop title pages for their stories. Although there is nothing to prevent the creation of title pages for *Story Tree* stories, there is no mention of title pages in the documentation and no examples provided on the disk. In the eyes of my nine-year old tester, "real" stories have title pages. Providing directions for their creation would be an easy and valuable addition to the program.

Overall Recommendation

In summary, I recommend Scholastic's *Story Tree* very highly! It is a well designed program that is educationally sound and fun for students to use. Furthermore, it is extremely versatile. It could find a comfortable home at many different grade levels, and teachers will discover interesting applications in a number of content areas. Because of its flexibility, it is appropriate for classrooms with a wide range in student ability and can be used quite successfully with special needs students. It is appropriate for students to use independently, but equally effective if used with students in pairs or in small groups.

Of equal importance to teachers should be the fact that students **like** *Story Tree,* and once the basics are learned will often spend their own time embellishing stories begun in class. This seems to be true even for populations of students traditionally turned off to writing. One of the teachers interviewed for this review related that he had several ninth grade boys come into his computer lab daily, on their own time, in order to write with *Story Tree.* These boys taught themselves the program and then spent approximately an hour a day, five days a week, for six to eight weeks writing and rewriting their stories. Any writing software that elicits that kind of commitment deserves to be looked at seriously!

Acknowledgement

The reviewer would like to thank the teachers of Eugene School District 4J who shared their students and their experiences using *Story Tree.* Special thanks go to Tom Layton, computer teacher at South Eugene High School, who graciously shared ideas arising from his use of *Story Tree* with students who have special learning needs and also provided many of the suggestions for improving the program.

Publisher's Reply: None.

© Copyright 1985 ICCE

Reviews

This month TCT's reviews include two well-known products that have been expanded or revised. With *Apple Logo II*, the larger memory and special features of the 128K Apple IIe and IIc are available for Logo programmers.

The *Bank Street Writer*, on the market since 1982, has recently been revised. There are now several versions of this popular word processor available through both Scholastic and Broderbund. Also reviewed this month, the *Bank Street Speller* provides several features that may be of particular interest to both students and teachers.

For primary-grade teachers new to Logo, a resource kit by Houghton Mifflin is available. *Introducing Logo to Primary Children* provides a classroom kit of teacher-developed instructional support materials.

Two French foreign-language programs are also featured. *Les Batisseurs d'Empire*, the French-language version of *Empire I: World Builders*, has been available in the U.S. for over a year. After examining several programs, this science fiction simulation game was selected by the reviewer as one of her favorites for third and fourth year French students. *Le Francais Par Ordinateur* offers three beginning French programs. Color graphics, music and a variety of learning activities introduce the student to French vocabulary and the culture.

BANK STREET WRITER

1984, Revised Apple Version
Personal use edition:
Broderbund Software, Inc., 17 Paul Drive, San Rafael, CA 94903
School use edition:
Scholastic Inc., 902 Sylvan Ave., Englewood Cliffs, NJ 07632.
(Includes three program disks, Student Guide, Teacher's Reference Guide)
Target Grades: 4-8
Subject: Word processing
Hardware Required:
Broderbund—
40-column version: Apple II+ (64K); Apple IIe (64K)
40/80-column version: Apple IIe (128K) Revision B board; Apple IIc
1 disk drive
Scholastic—
40-column version: Apple IIe (64K)
40/80-column version: Apple IIc (128K); Apple IIc
1 disk drive
Cost: $69.95 (Broderbund); $95 (Scholastic)
Reviewed by: Keith Wetzel, Computers in Education, University of Oregon

When the first *Bank Street Writer (BSW)* came out, many of us were caught by a vision of the promise it held for writing in the elementary and middle schools. Since that time I have closely observed students using word processors in the intermediate grades, and my euphoria has been tempered by classroom realities. Accordingly, this review concentrates on the new *BSW*, but it also reflects new knowledge about the use of word processors by students in grades four through eight.

As a writing teacher, my primary criterion for any word processor is that it allows students to move easily and efficiently between the different stages of the writing process. Writing is not a linear, lock-step process of pre-writing, composing and revising. Rather the process is recursive: The writer moves back and forth between pre-writing, composing and revising. This process continues until the writer is satisfied with the product.

The new versions of *BSW* for the Apple IIe and IIc, marketed by Scholastic and Broderbund, are substantially different from the original *BSW*. Each now has a free-moving cursor which may be moved around the screen in the WRITE mode to make insertions and deletions. ***This feature makes the new version significantly better than the original version.*** In addition, it satisfies the evaluation criterion, allowing students to move easily and efficiently between the stages of writing.

The original *BSW* and the updated Broderbund version when used on the Apple II+ (64K) hamper revision by adopting a two-mode system: one mode for entering text and another for editing. For example, seven different steps are required to jump up two paragraphs, add a new sentence and return to continue writing. It is much easier for a student to use a word processor which allows text to be edited as it is written. For this reason, *BSW* in any edition for the Apple II+ is clearly less desirable.

Apple IIe (64K) Features

BSW for the Apple IIe (128K) and IIc can do everything the IIe (64K) version can do, and more. The Apple IIe (64K) version has these new features:

- Full cursor movement on the write screen
- Printer format commands (boldface, underline, etc.)
- Optional mouse control to select functions or scroll text
- Help screen
- End-of-line character
- New-page command

Apple IIe (128K) and IIc Features

Additional features available only on the Apple IIe (128K) and IIc versions are:

- Choice of anywhere from 40- to 80-column display
- Calculator for solving basic math

Features Missing

The *BSW* developers have included some new features to make this word processing program more powerful, and other features have been left out. One missing feature is that the screen display does not necessarily show the printed format; that is, what you see is *not necessarily* what you get. With some word processing programs what you see on the screen is exactly what is printed onto paper.

Missing also is a straightforward way of single spacing portions of text in a double-spaced report.

Documentation

The documentation for the Broderbund versions and the teacher's guide for the Scholastic version are clear and helpful.

Discussion

Developers of the new *BSW* have had to make decisions concerning what new features to include. Questions needing to be addressed include: "What do students in grades four through eight need in order to write more efficiently?" "What features are needed to allow *BSW* to keep up with the features of other word processors?"

The first question is clearly answered. The decision to allow free cursor movement in the WRITE mode certainly makes the new *BSW* a more powerful tool for writing. The other question must be answered individually by the purchaser, on the basis of those features needed for his/her purposes.

© Copyright 1985 ICCE

Composition Teachers

Teachers who want students to use any word processor (including *BSW*) in learning composition need to fully understand what word processors actually do. Recently a fourth-grade teacher told me, "I still have to teach my entire writing program. What is really important is what the teacher *teaches* about writing. It is not simply the computer." This reflects an important point: Word processors do not teach writing. They do not tutor students or substitute for a writing program. The word processor, rather, is a tool in the writing process.

My second caution concerns expectations: It is commonly assumed that students will write more if they can revise without recopying by hand. Before jumping to this conclusion, consider carefully whether your students can type adequately. Without this prerequisite skill, students may not write as much with a word processor as they do with paper and pencil.

Recommendations

My review of the original *BSW* appeared in the August 1983 *TCT* (Vol. 11, No. 1), when it was the only word processor designed for fourth through eighth graders that I knew of. Now there are other word processors designed specifically for students, as well as adult word processors which may be considered for students because they are not difficult to learn or use.

It was also easier for me to be enthusiastic before realizing the importance of adequate keyboarding skills and acquiring a thorough knowledge of the teaching of composition. Now word processors have been in classrooms long enough that we can "look before we leap."

Consequently, I advise that you carefully examine the idea of using word processors in composition, and if you decide to take the leap, use several word processors yourself before making a decision. Despite my cautionary tone, I find the new *BSW* is one of the top word processors available for students in grades four through eight, and I recommend you include it on your list of word processors to be examined.

Publisher's Reply

Thanks very much for sending me the review copy for the new *Bank Street Writer*. Generally, I found the review to be fair and accurate. But I did want to point out a couple of things:

The reviewer mentions that "the screen display does not necessarily show the printed format." While this is technically true, it is possible to get, quite quickly and easily, printed copies that mirror what the screen shows—and vice versa. If you are using the 40-column version, selecting "Print-Draft" automatically gives you a printout that uses the same line length as the document shown on the screen. If you are using the 80-column version, an option in the Utility Program allows you to set the screen display for 40 columns, 80 columns, or any line width between 40 and 80 columns. As a result, once you decide on the line width that you want to use in your printed copy, you can set the screen for the same display.

The Scholastic school edition of the new *Bank Street Writer* includes, on the same disk, versions that will work with the 64K Apple IIe, the 128K IIe and the IIc. This is an important feature for schools that have a mix of 64K and 128K IIe's, for schools that currently have IIe's and are planning to upgrade them with an extended 80-column card, and schools with IIe's that plan to purchase IIc's.

The reviewer also remarks that schools will have to determine for themselves which of the other enhanced features meet their needs. While this is very true, I think we've done a good job of developing and testing features that make the *Bank Street Writer* an even better product for both students and teachers. As we mention in the documentation, while the new *BSW* is just as easy to use as the original version, the expanded edition allows students to grow with the program, gradually uncovering new features as they become comfortable with the more basic functions.

Philip Miller
Senior Project Editor
Scholastic Software

BANK STREET SPELLER (1984)

Personal use edition:
Broderbund Software, Inc., 17 Paul Drive, San Rafael, CA 94903
School use edition:
Scholastic, Inc., 902 Sylvan Ave., Englewood Cliffs, NJ 07632
Subject: Spelling checker
Hardware and Software Required: 48K Apple II+/IIe/IIc
One disk drive (two required for adding words to the dictionary)
Bank Street Writer word processor
Cost: $69.95 (Broderbund); $64.95 (Scholastic)
Reviewed by: Doris Carey, Computers in Education, University of Oregon

Word processor users have definite ideas about what spelling checkers should do. They should, of course, point out spelling errors, and in addition should refrain from correcting these errors until instructed to do so, in case the word has been deliberately misspelled or it is a proper noun or foreign word that does not appear in the standard English dictionary. *Bank Street Speller* accomplishes these objectives by comparing each word in the user's text file to its 30,000-word dictionary (derived from the *Random House Dictionary,* Concise Edition) and allowing the user the opportunity to decide whether to replace the word or leave it intact. *Bank Street Speller* can't guarantee the correct spelling of words that are not in its dictionary, nor can it point out errors in style or grammar—but neither can a regular dictionary.

Bank Street Speller provides several additional features that may be of particular interest to teachers: an alphabetic list of words used with frequencies; the total count of words used; and a count of *different* words. My small sample file, for example, used a total of 52 words, but only 45 different words, since "with" was used twice and the words "a," "the" and "to" were each used three times. The list of different words with their frequencies can be printed out or viewed on screen. The totals appear only on the screen. With a bit of imagination, a teacher can use this feature for games and activities that enhance students' awareness of their vocabularies. Students might be asked to draw a histogram that illustrates their use of two-letter, three-letter, four-letter, etc., words. They might also make a game of writing a coherent paragraph in which no word is used more than once. The user can also quickly alphabetize a list of words.

Bank Street Speller will also print out a list of words that it did not find in its dictionary. Again, students can make a game of picking out words that are probably misspelled, as opposed to exotic words or proper nouns that should be retained. Teachers can decide what to do about slang, since words like "ain't" do not appear in the dictionary. Words categorized as "obscene" in my *American Heritage Dictionary* also seem to have been left out of *Bank Street Speller.* The **Make New Dictionary**

© Copyright 1985 ICCE

feature allows the user to add or delete words to the dictionary and to create new dictionaries. I see great possibilities here for foreign language teachers.

Bank Street Speller's **Search Dictionary** feature is particularly interesting. It prints a list of words in the dictionary that *almost* match a particular misspelled word. For example, I included "woud" in my text file, and a search of the dictionary yielded the words "wood," "would," "word" and "wound" as suggested replacements. I was also able to check the spelling of "separate" by entering "sep?rate." Entering "r=d" gave me a list of all words in the dictionary that begin with "r" and end with "d." The question mark allows for a single character, whereas the equals sign is a wild card that allows for the substitution of any number of characters.

After I finished making changes to my text file, I decided to retain 14 of the 18 words that *Bank Street Speller* had not found in its dictionary. The four misspelled words that I did replace were printed out for me in this form:

woud ⟶ would

As a teacher, I was happy to see that students could keep a list of these corrections in a scrapbook or notebook for future reference or review.

Bank Street Speller is easy to use. Like its parent word processor, *Bank Street Writer*, the program's menus are always visible at the top of the screen, with questions and prompts appearing on screen when needed. As an experienced user of *Bank Street Writer*, I was able to fully check, edit and print out my file in 10 minutes the first time I attempted to use *Speller*. This included the time it took to obtain printouts of my word lists, errors and corrections. I spent another minute or two accessing the utility section to customize the program to accept a keyboard modification for upper and lower case letters (since I was using an Apple II+).

Teachers who plan to purchase this program should be aware that single-drive systems will require a vast number of disk changes. Once *Speller* was loaded, I had to load my file from my data disk, then alternate between the dictionary and data disk each time I needed to search the dictionary. Without these searches, I needed at least five disk changes from beginning to end, including reloading *Bank Street Writer* and the data file to print out the final copy. I would hastily add that these annoyances are not due to a design flaw, but to the limits of the technology. It would be desirable to have both programs in memory simultaneously, but the obvious tradeoff would be a requirement of more than 48K of available memory.

One small design decision that I would change is the arrangement of word lists. They are alphabetized, but appear in two columns arranged to be read left to right; I would prefer to scan down a column instead of across rows.

I would strongly recommend the purchase of *Bank Street Speller*, not only because it fulfills its objectives, but because its many features provide an environment for experimentation and creative play. I expect it won't be long before an enterprising teacher compiles a list of activities for use with this program—I wish I had the time to do it myself.

© Copyright 1985 ICCE

Doris Carey

The Bank Street Writer
Personal Use: © 1982 from Broderbund Software, Inc., 1938 Fourth St., San Rafael, CA 94901. Cost: $62.50.
School Use: © 1983 from Scholastic, Inc., 902 Sylvan Ave., Englewood Cliffs, NJ 07632. Cost: $95.00.
Grades 4-12/Word Processing.
Hardware Description: Apple II, Apple IIe, Atari. (Commodore 64 being developed.)
Reviewed by Keith Wetzel, University of Oregon, Eugene, Oregon.

The *Bank Street Writer* (BSW) is a word processor designed for students. It represents a good use of the computer in an upper elementary or junior high school classroom.

Typically word processors have been adult-oriented. They require days to learn to operate, thus they are not accessible to students. In contrast, the BSW requires only one to two hours to learn to use. No knowledge of programming or math is necessary to use it.

The BSW potentially enables a classroom teacher to use the ideas of process writing—prewriting, writing and rewriting—efficiently in the classroom. One of the keys to improving student writing is to encourage students to write more. A word processor appropriate for students is motivating, and reduces the drudgery associated with editing and recopying.

How It Works

The BSW is organized like a three-page menu. The first page is the writing or text-entering page. The user begins here. The words ENTER TEXT are always highlighted at the top of the page. The first page allows the word processor to be used much as a typewriter. The move to the EDIT page is made by pressing the ESC (escape) key, with the words WRITE OR CORRECT highlighted at the top of the screen on this page.

The philosophy of the BSW's authors is evident. They wanted one mode for entering text. After the user completes entering all his/her ideas, then it is time to move to another distinct mode to correct errors, edit sentences or move ideas around.

The editing page has this menu at the top:

ERASE MOVE FIND TRANSFER
UNERASE MOVEBACK REPLACE

The user chooses the correct command by using the forward and backward arrows. When the correct choice is highlighted, the user presses the return key and follows the directions which appear at the top of the screen. The commands do what their names suggest. The user does not need to remember them because they are always present at the top of the screen.

The third page of the menu is the TRANSFER text page. This page allows the user to save the typed text. Then the text can be printed on paper, deleted from memory or retrieved from memory. Once again the choices are listed at the top of the screen.

There are error traps to prevent mistakes. The user is asked, "Are you sure you want to . . .?"

Apple II vs. Apple IIe

This word processor has upper and lower case letter capability on both Apples. The Apple II+ requires the user to press two keys (the shift key and the N key) to print a capital letter. A minor modification to the Apple II+ enables the user to type a capital letter using only the shift keys. The Apple IIe requires no modification to type capital letters in the usual format. The IIe also has a shift-lock feature which enables the user to type continuous capital letters.

The BSW can be used on either Apple. The same piece of software will operate on both machines. The Apple II+ uses the I J K M keys to move the cursor in the text edit mode. The Apple IIe has separate arrow keys ←↑↓→ to accomplish the same task. The BSW works appropriately with either machine, and the tutorial also reflects these differences.

Printed Copy

The BSW uses any standard printer and requires no special set-up. To print copies the user is required to follow the directions after highlighting TRANSFER and then PRINT FINAL. A series of questions appear which allow the user to adjust the number of characters printed per line, the spacing (single, double or triple), the numbering of each page and other printing details. After each copy the user will be asked if another copy is desired.

Limitations

There are few limitations of the BSW for intermediate and junior high age students. One limita-

© Copyright 1985 ICCE

tion might be the difference between what appears on the screen and what appears on the printed copy. The BSW allows forty characters to be printed per line on the screen. The number of characters printed per line are adjusted prior to printing. The standard typing format is about sixty-five characters per line. Consequently, the printed text does not appear in identical form to the text on the screen. This does not seem to cause students any problems.

Broderbund and Scholastic

There are two versions of the BSW—they differ primarily in the audience they are directed towards, and consequently the way they are packaged and marketed. Broderbund distributes the program only to individuals, their market aimed primarily at home users. There is no preview or review period prior to purchase. It includes two disks and a small reference guide at a cost of $62.50. Scholastic distributes the program only to schools. It includes three disks and a teacher's manual. The tutorial side of the disk may be copied if you wish to use the tutorial with groups. Schools are allowed a thirty-day preview period. The cost is $95.00.

Documentation or Teacher's Guide

Broderbund's user manual is concise. In combination with the tutorial, which is on the opposite side of the disk, it answers questions and gives the user practice in entering and editing text.

The Scholastic version shares the same software. Though the word processor is the same and the tutorial is the same, the documentation is different, as it is designed to meet the needs of teachers and students. It includes a teacher reference guide, a student reference guide, a question and answer section and a student activities section. The student reference material can be copied and distributed to students.

The BSW is not easy for students in intermediate grades to learn to use—guidance is necessary for quick, efficient learning. I wanted to find out how easily a seventh grader would learn to use the BSW with only the assistance of the reference material. In forty-five minutes, and with a few questions, this student learned to use it.

How does that compare with another seventh grader using only the tutorial? The tutorial beeped often, indicating a student error. When the student made an error, the tutorial did not offer hints or an analysis of the error. The consistent beeps and the message, "You did not follow directions," frustrated the student. The student's comments upon completion of the tutorial in fifty minutes was, "It was tedious, but I did learn something." The student who used only the reference material and asked a few questions had a more positive experience.

Several intermediate grade level teachers also commented that the tutorial was not appropriate for their students. Their reasons were similar to the problems discussed earlier.

The BSW may not be understandable to the student user at first, but a judicious use of the reference material, the student activities, a teacher's oral introduction and the tutorial would seem to be an effective way to allow students to use it independently after a few hours of practice.

Rating

Both versions—Broderbund and Scholastic—contain an excellent word processor which meets a real need. The Broderbund streamlined package is fine for home use. And the Scholastic package with an additional disk and a teacher's manual is appropriate for school use. Why is the BSW excellent? First, it is an educational piece of software which does what it was intended to do—make an advanced writing tool accessible to parents, students and teachers. Second, it compares favorably in price to other commercial word processors. Third, the students who I questioned about rewriting a composition by hand or editing it using the word processor emphatically endorsed the word processor. A fifth/sixth grade teacher who uses BSW consistently for language arts activities summed it up: "I like it, but the kids *love* it."

Broderbund Distributors:

UK: Tele Swift, 189 London Rd., North End, Portsmouth, England.

Australia: Imagineering, P.O. Box 4601, Sydney, Australia, NSW 2001.

Report Card (1983)
Sensible Software, 6619 Perham Drive, West Bloomfield, Michigan 48033
Hardware Description: 48K Apple, Disk Drive (3.3 DOS), Applesoft BASIC, printer.
Cost: $59.95
Reviewed by Sam Miller, Eugene School District 4J, Eugene, Oregon.

Report Card is a grading system organized around three basic grading operations: (1) entering and editing data; (2) displaying data; and (3) printing data. The program computes grade averages and percentages and assigns letter grades on the basis of percentages specified by the user.

Report Card holds data for up to 300 students and an unlimited number of data disks may be established.

The program includes the option to sort and print student data either individually or by a class. Each class may include up to 50 graded activities. *Report Card* will also display or print the results of selected activities such as quizzes and midtern examinations for a particular student or an entire class. Data can be sorted alphabetically, by grade, or by percentage score.

© Copyright 1985 ICCE

The various functions of the *Report Card* program are accessed through a series of menus. Each menu deals with a specific type of action such as entering student data or printing information. First-time users may experience some initial frustration using a system of embedded menus. However, I found that with an hour's worth of practice, the entire structure of *Report Card* became easy to manipulate.

The documentation is written with a limited amount of technical jargon and provides an introductory tutorial. The only negative aspect of the documentation is that the very small typeface is difficult to read. It would also be helpful if an index were included.

A few flaws exist in the program, but they are relatively minor. For instance, the program limits the number of letters that may be used to describe an activity to 15 letters. There are occasions when an activity name will be longer than 15 letters and it's a nuisance to have to remember what abbreviated descriptions stand for. For educators who believe in sending home frequent progress reports it would be helpful if the program allowed the user to specify some brief comments pertinent to a student's grades. Grade reports printed out by the program are a bit sterile in appearance without accompanying remarks.

Report Card has excellent on-screen prompts that assist the user by asking easy-to-understand questions and specifying the correct response options.

The program provides several features that make me want to dance in the street. These features include: (1) exceptional error-trapping that prevents the user from misassigning letter grades to a specific percentage range; (2) extremely easy editing functions that allow for spelling/grade corrections; (3) provisions for making backup copies of data easily; (4) provisions for initializing a data disk; (5) assignment of a previously constructed class roster to a new activity; and (6) the option to assign identification numbers to students for the purpose of security/anonymity.

All teachers face the laborious task of trying to keep accurate records and providing students with information about their progress. *Report Card* is an excellent example of software that fills a definite need in education. I intend to use it next year when I am faced with the task of keeping records for over 80 middle school students.

Distributor:

UK: S.B.D. Software, Unit 4, South Downwork, Osiers Road, London, England 5W18 1NL.

© Copyright 1985 ICCE

Reviews

APPLEWORKS

Apple Computer, Inc.
Subject: Integrated applications program (word processor, data base, spreadsheet).
Hardware Required: Apple IIe, 80-column Text Card; Apple IIc; 64K or 128K Ram (recommended); 1 disk drive (2 recommended).
Cost: $250.
Reviewed by: Jack Turner, Computer Coordinator, District 4J, Eugene, OR.

"Yes, But Can It Climb Trees, Too?"

Imagine a school where the principal, while observing a class, could help a student set up a data base record format for a project on gas mileage as a function of engine displacement. Or where a pupil could help a teacher set up a spreadsheet template to track student progress. Or where a social studies class could create a data-based unit on the pioneers who travelled the Oregon Trail.

We now have access to a computer program—*AppleWorks*—which could benefit every participant in schools from the custodian to the principal. This program could change substantially the nature of education if its full potential were implemented by teachers, students, administrators and classified staff.

AppleWorks, an integrated tool program sold by Apple Computer, is indeed an amazing product. This $250 package combines a word processor, a data base and a spreadsheet program into a very useful productivity tool kit which is impressive, both in scope and simplicity. Although the price may appear a bit steep, it is in fact a very reasonable alternative to purchasing the three applications programs separately. You are unlikely to find any combined price total of less than $350 for a separate data base program, word processor and spreadsheet of any quality. And if you *could* find three programs at bargain prices, you would be confronted with learning three different sets of control commands. Not so with *AppleWorks*, whose three central programs all share (with very few exceptions) the same commands.

A second and even more important aspect of integration among the programs is their ability to share files. No more trying to outguess your word processor, wondering if you've left adequate text space for a table of figures generated by your spreadsheet. Now you can simply leave a marker in the text and tell the program to take care of the cutting and pasting.

AppleWorks runs with ProDOS on either the Apple IIe or IIc and while it can function with a single disk drive, you will soon want a second drive. Its capabilities are severely limited if the programs find only 64K to operate with. Maximum file length within the word processor at 64K is approximately eight pages of text; at 128K the capacity more than triples. Similarly, the data base at 64K can hold only about 140 average-sized records, while 128K will manage approximately 700. Obviously, *AppleWorks* will be perfect for the IIc with its built-in 80-column, extended memory. But even the IIc owner will soon be shopping for a second disk drive.

The program is not copy-protected; the user is instructed to make a duplicate copy from which to work. The copyright text inside the cover of the Reference Manual implies that Apple will be amenable to multi-use licensing for shared-disk systems. Inquiries can be directed to Apple Computer, Inc.

The fact that *AppleWorks* uses a memory-based architecture for its data base program allows for very rapid data manipulation, but the trade-off might seriously limit the program's value for applications like student-record keeping in a large school. Although *AppleWorks* will copy easily to Apple's ProFile hard disk, this additional five megabytes of storage capacity will not increase the number of records you can handle within a single file. The hard disk as a secondary storage device will allow you more rapid access to a larger number of data files (or text documents or spreadsheets), but it will have no effect on the maximum size allowable for each. Unfortunately, that is a function of RAM and not disk capacity. (A reminder about *AppleWorks*' primary reliance on RAM storage—back up your work frequently. *AppleWorks* makes it very easy.)

Two more criticisms that derive from system and software architecture should leaven my very favorable impressions of *AppleWorks*. It is a nuisance to have to start up the system with a separate Start-up Disk, the source of the operating system instructions for ProDOS. After an eleven-second load time, you are instructed to insert the main Program Disk which holds the three tool programs. It would have been more convenient if a single disk could contain everything needed to get you into the program, but *AppleWorks* alone is so large that there isn't enough room to include the ProDOS instructions. The other, and more serious, shortcoming is that *AppleWorks* has no true mail merge capability, at least not an automated merger. The word processor does have a feature which will cue the user to intervene and type personalized information into printed output directly from the keyboard. But it is tiresome to be interrupted and called back to the computer every three minutes as the printer pauses to wait for you to type the recipient's name into your form letter. For some users this omitted feature is a fatal flaw—particularly since the word processor and data base components are so nicely integrated in other respects. This omission also can be attributed to the scarcity of available space on the main program disk.

Haba Systems to the rescue! This software house recently announced *Haba Merge*, a companion program for *AppleWorks* which will fill the mail merge gap and throw in custom labels for mailing and other purposes. Suggested price for this new software is $69.95. It is not clear whether *Haba Merge* will print mailing labels in a format wider than "one up." While I found making mailing labels straightforward on *AppleWorks*, it can only provide the "one up" format, i.e., a single column down the sheet rather than up to five rows of gummed labels across. Haba Systems had an important hand in the creation of *AppleWorks*, so *Haba Merge* is formatted identically.

This section will look at the three main applications programs plus the clipboard where you put things which are to be moved around. Space limitations prevent this review from providing extensive detail on program operation—thorough reviews of each component could easily qualify as separate reviews. Suffice it to say that it took Apple's documentation writers over 450 pages of tutorial and reference text combined with an exceptional disk-based introduction ("Apple Presents *AppleWorks*") to provide clear user directions. While the sheer volume of documentation provided—two books, a four-page quick reference card, an interactive, two-sided disk tutorial and a disk of sample files—seems at first oppressive, it is a model of

© Copyright 1985 ICCE

clarity and organization.

The Word Processor

It is clean and easy to use. Except for right margin justification, formatting is done on screen from within the text editor, so the screen is usually a literal representation of what the print copy will look like. An excellent feature allows you to execute fast and literal screen dumps to the printer from anywhere within any *AppleWorks* component. That means you won't have to wonder what that report format, made and saved six months ago under the file name "Spiffy," looks like. With *AppleWorks* you can create a format, "photograph it" to the printer with two keystrokes, and return to where you left off.

Almost all control commands and printer options use the Open Apple key in combination with one mnemonically chosen letter key; e.g., to SAVE to disk you use Open Apple-S. I was pleased at how quickly these codes felt natural to use. Even better is the fact that all such code schemes are confined to two keystrokes. The word processor I used to use required five sequenced keystrokes to invoke emphasized print.

One minor criticism with the word processor is that, except to get underline or boldface functions, you have to leave the text editor and go into the Print Options menu to make each formatting decision. For example, if you wish to center a title there appears no way to do so unless you type Open Apple-O which instantaneously fills the bottom half of the screen with options, select CN as the option needed to center the text, hit ESC to return to the text editor so that you can reposition the cursor to where the centering should end, return to printer options with another Open Apple-O, select UJ (stands for "Unjustified") and then hit ESC again to return to the editor and continue text entry. Although the description takes longer than the act, it is clumsy, particularly if you want to make a number of formatting changes in a row.

It could be that the speed you lose going in and out of the Print Options menu while composing/formatting is repaid when you save a file to disk. Thanks to ProDOS, saving even long files is much quicker than what DOS 3.3 led us to expect.

The Data Base

If you have used Quick File IIe, an earlier stand-alone data base developed by Rupert Lissner, you'll recognize the *AppleWorks* data base instantly. Lissner, who wrote all of *AppleWorks*, polished up his earlier data base for inclusion. I hope it becomes a new standard for data base programmers—it is simply superb. I have learned five data base programs in as many years; this one ends my search. Files are designed by typing up to 30 field names (20 characters maximum) and then entering your data. Maximum field length is limited to the space remaining on an eight-column line to the right of the field name. If you wish to build a data base from data already contained elsewhere, *AppleWorks* will accept it from ASCII, DIF, VisiCalc or Quick File IIe files.

The data base allows you versatility in browsing through records and in developing report formats. To view multiple records simultaneously you simply zoom (with Open Apple-Z) into the multiple-record format. If a particular record needs your editing attention, you can then zoom in on it as an expanded single-record layout—a unique feature among data bases.

The two main reporting formats are characterized as "tables" or "labels." "Tables" has printed output resembling the columnar arrangement seen on the screen when in multiple record format. "Labels" arranges information from data fields in the form typified by mailing labels; successive fields are beneath, rather than beside, each other. Perhaps not unique, but a very well-done implementation.

My only regret within the data base is that you cannot tell the program to simply repeat, or copy, information from a given field in the last record into the same field in the record you are currently typing. This missing feature would spare us many repetitious entries and speed up data loading. *AppleWorks* chooses instead to let you develop or change Standard Values, where you name the value to be inserted into one or more fields for *all* new records. The effect is the same, but the user has less flexibility.

The Spreadsheet

The maximum theoretical size of the spreadsheet matrix is 999 rows deep and 127 columns wide—an incredible 126,873 empty cells. When you begin putting information inside the cells, the practical limits shrink to approximately 6,000 cells in a 128K machine. It probably won't hold the entire U.S. budget document, but it should be adequate.

The spreadsheet has all standard math operations typically found in this type of program (no trigonometry, however) and is easy to learn. If you use the other two programs first, you will already know many of the commands. *AppleWorks* also adds some unique features that make its spreadsheet easy to learn and use. With the 80-column card's highlighting capability, the spreadsheet highlights cells you wish to move, copy, change in width or delete. The program can sort and find labels or values and can type labels that automatically extend into the following columns; it has protected cells and can split the screen either vertically or horizontally to permit simultaneous viewing of separate portions of the worksheet.

The spreadsheet has a number of unexpected features and only the order of recalculation is a bad surprise. If a cell is out of order, it will have its formula recalculated with old (and usually incorrect) values. Consequently, a bottom-line figure must follow the cells upon which it depends or the information will not be accurate. The Tutorial Manual acknowledges this problem and has a reasonable explanation for how to avoid it.

Summary

AppleWorks may not be an improvement on *Lotus 1-2-3*, but Rupert Lissner has created an exciting shift given Apple's CPU and RAM limitations. The programs work well, both separately and together. The integration is not as complete as a theoretician might demand; in addition to having no true mail merge capability, you cannot retrieve information from the records within a data base and have them transferred to your spreadsheet. I can live with that. The combined set of programs exhibits so many thoughtful ideas that it would be difficult to list them all without sounding like a testimonial.

In the best of all worlds, *AppleWorks* would be included with every 128K computer. Until that time comes, you'll need to make it happen by yourself. Unless you have very specialized needs, this program will become your most versatile computer tool.

© Copyright 1985 ICCE

Reviews

Linda Rathje, Editor

Musicland
Rock 'N' Rhythm
MusicWorks
Bank Street MusicWriter
Operation: Frog

TESS '85

MUSICLAND

Freelance Musicworks, Inc., 18 Haviland Street, Boston, MA 02115, 617/536-9494

Target Age Group: Child to adult

Subject: Music

Hardware Required: Apple II, II+ or IIe (64K); 1 disk drive, color TV or monitor, KoalaPad or joystick (preferably not spring-loaded), Mountain Computer Music System synthesizer cards, headphones and Y-cable or stereo amplifier

Cost: $99.95 (current version); new version (available June 1985) $100-150. Disks are copy protected. Back-up copies are $20; old versions may be upgraded for $20 by sending in old copy, or proof of purchase.

Reviewed by: Jim Ellickson-Brown, Elementary Music Specialist, Mountain View School, Corvallis, OR

Introduction

Musicland is a set of four interrelated games for exploring music developed by Martin Lamb of the University of Toronto Computer Systems Research Group. Lamb titled his program after the imaginary Mathland described by Seymour Papert in *Mindstorms,* in which the developer of Logo described a place where children could learn mathematical concepts as easily as they would learn French by living in France. *Musicland* is designed to give the user equally immediate access to the fundamental principles and expressive power of music.

"MUSICLAND is a perfect example of the realization of the idea of a learning environment with no threshold and no ceiling."

Description

The user experiments with music using four games:

In MUSIC DOODLES shapes are drawn on a music staff on the screen using the joystick or KoalaPad. These shapes are turned into notes by the program, and you can hear what you have drawn. Selected parts of these musical shapes can be moved about on the screen, stretched, squeezed and turned inside-out. You may listen to these transformed shapes at any point and plant copies of them to build complex compositions; related shapes are color-coded to show their source. The basic musical idea explored in this game is the concept of musical ideas as gestures or lines. Among the techniques explored in transforming the shapes are: transposition (moving an idea higher and lower), diminution and augmentation (shrinking and stretching a musical idea), as well as retrogradation and inversion (mirror images of a musical idea in the rhythm and pitch dimensions, respectively, created in *Musicland* by shrinking a shape past the zero point).

In TIMBRE PAINTING you give the notes of your composition different tone qualities by painting them with any of five colors. In formal music study, this is the concept of orchestration.

The designs you have made in MUSIC DOODLES appear as blocks of music in MUSIC BLOCKS. You arrange the blocks to make compositions using repetition and contrast.

In SOUND FACTORY you explore and change the tone qualities of the col-

In Musicland, you explore music by making pictures. The pictures and patterns become melodies and the colors determine the tones of the melodies.

© Copyright 1985 ICCE

ors you used in TIMBRE PAINTING, learning about the physics of sound in the process. The tone colors created may be taken back to be used in TIMBRE PAINTING.

Included in the package is a 25-page manual and a set of activity cards for each game, including some suggestions for integrating *Musicland* with studies in language arts, mathematics and science.

Using the Program

The primary input device in *Musicland* is the KoalaPad or joystick; the user is required to use the keyboard only occasionally. The first of these instances is the main menu, where you may see a description of the games, specify whether you are using a joystick or KoalaPad, reset the original blocks and tone colors which come with the program, or go to one of the games. You may return to the main menu from any game by pressing the ESC key and confirming your decision.

Once inside a game the input device controls a cross-shaped cursor on the high-resolution graphics screen; choices are made by pointing to words or icons on the screen and pushing one of the buttons. Arriving in MUSIC DOODLES, for example, you will see a grand staff in the center of the screen; positioning the cursor on the word DRAW above the staff and clicking one of the buttons will turn the cursor into a pencil, ready for use. To draw a line, you position the pen with the joystick or KoalaPad and hold down a button as you move. At this point the additional options PLAY, UNDO, CLEAR, SAVE and TRANSFORM also appear, allowing you to hear your work, erase a mistake, begin anew, save a piece and begin to develop an idea. You are also given the options of viewing a help screen, restoring a saved piece, going to TIMBRE PAINTING to orchestrate your work, or taking your composition to MUSIC BLOCKS.

Choosing the TRANSFORM option turns the cursor into an arrow. You select notes to transform by painting over them; pointing to DONE and pushing a button completes the selection. Bringing the cursor onto the staff you will see a copy of the notes you selected. If you move higher and lower on the staff, they move with you; press the first button and you will hear what the notes sound like in a particular spot. Pushing the second button will plant a copy of the shape; it will become colored and remain. You may plant many copies of the same shape and even erase the original. Stretching, shrinking and turning ideas inside-out is done using the keyboard. When you want to hear your work, selecting DONE will return you to the main MUSIC DOODLES screen where you can play your work. If you are satisfied with your work, you may begin a new cycle of transformations; if not, select UNDO to restore your composition as it was, including an erased original.

Once you have constructed a piece, you may take it with you to TIMBRE PAINTING. Selecting one of the five colored squares at the top of the screen will let you hear the tone quality associated with each; the cursor has now become a paintbrush. Moving onto the staff, push a button to begin painting. You do not have to cover notes completely and do not need to worry about being messy. When you select PLAY, the computer will clean up the screen and let you hear the result. You may then choose another color and continue painting. At this point you may wish to go to MUSIC BLOCKS, either via the main menu using the ESC key or by way of MUSIC DOODLES if you wish to take your work with you.

Arriving in MUSIC BLOCKS, you will see on the bottom of the screen five squares, each a miniature version of a music doodle. Above these is a blank area with room for 15 blocks. You may listen to a block before selecting it; choosing a block will place it in the next available free space. You may hear the arrangement you have created when you wish; each block lights up as it plays. A block may also be dragged to a new place, gotten rid of (by dragging it off the screen) and inserted; UNDO, CLEAR, SAVE and RESTORE options are also available. Included with *Musicland* is an original set of five blocks: a smiling face, a fragment of a ragtime song, and three blocks which may be arranged to play a familiar children's song.

The fourth game, SOUND FACTORY, allows you to experiment with the timbre or tone color of a note. If you produce something you like, you may take it with you to TIMBRE PAINTING. You use the KoalaPad or joystick to adjust the volume and pitch, and control the amount of 16 different harmonics or overtones. You may also examine the resultant waveform, draw your own freehand, and make adjustments in the envelope (the volume of a sound from beginning to end) and overall length of the sound. When *Musicland* plays a note, it actually plays two notes at once; you may offset these slightly to produce an effect similar to that of two instrumentalists playing together.

Strengths

Musicland is exceptionally well designed and easy to use. The documentation describes the games in detail and provides suggestions about introducing them to students; the set of learning activity cards includes guidelines for their use based on age group, ability level and experience with the games. In keeping with the intent of the program, these activities provide opportunities to explore both the structure and emotional content of music.

Musicland has been a valuable addition to my elementary school classroom. Younger children invariably grinned as they painted over the notes in TIMBRE PAINTING; when I presented an arrangement of four familiar tunes to some of my older students, they showed exceptional care in orchestrating the piece to bring out each tune. Third grade students spontaneously clapped when one student correctly arranged a set of MUSIC BLOCKS to form a song the group had just learned. A fourth grade student who struggles to keep up with many traditional activities of the class repeatedly sought out extra time to use the program. And when one of the most able students in the class also asked for extra time with the program, the two ended up working together to master the entire set of games without any help from me.

These examples are an indication of the role *Musicland* might play in teaching and motivating students of widely differing backgrounds and abilities in elementary and middle school, and beyond. *Musicland* could also serve as a valuable outlet for musical expression for the handicapped student.

Almost everyone responds to and enjoys music, but the difficulties of mastering instrumental technique and understanding musical notation often inhibit the development of this innate musicality. Many computer music programs compound the problem by requiring use of the typewriter keyboard and computer jargon. *Musicland* breaks down all of these barriers, encouraging musical self-expression and exploration, which can in turn develop the desire to understand and master instrumental technique and traditional musical notation.

Weaknesses

As impressive as *Musicland* is, it has four important shortcomings:

© Copyright 1985 ICCE

The first is the lack of choice of musical scales. A musician often thinks of composing a musical gesture or line based on a limited set of notes; this choice profoundly affects the sound of a musical line and is an important compositional decision. In *Musicland,* the musical lines drawn in MUSIC DOODLES are interpreted in terms of the complete set of notes or chromatic scale. Although I realize the value of having all 12 notes to work with, and know I can make adjustments by transforming individual notes upward or downward, the feeling of immediacy is not the same. Allowing the user to choose from or construct various scales would have added additional power to the program; imagine exploring the effect of the same drawing played in several different scale systems.

The second problem is the lack of control of tempo. The ability to change the overall speed at which a piece is played would not only add increased musical interest, but would also make it possible to adjust to students' ability to follow a given score. The ability to step through a piece note by note would also be valuable.

The third problem is the absence of a file management system. Only one student's work may be stored at a time on a disk. I have had several students create excellent compositions only to lose them permanently when another student's turn came.

The fourth problem is the limited set of sample compositions which come with the program. The most obvious need is for more pieces to be used in MUSIC BLOCKS and TIMBRE PAINTING. Additional sets of tone colors for use and study would also be of value.

Technical Details

The KoalaPad is far superior to a joystick for use with *Musicland*—the program was designed with a graphics tablet in mind. Oddly enough, each time you boot the program it assumes you are using a joystick unless you specify otherwise. If you forget to do so, you will notice the cursor jumping to the upper lefthand corner of the screen each time you lift the stylus. This will create unwanted lines on the screen if you are holding the button down at the time; it may also result in an accidental selection of the PLAY option. When the program expects a KoalaPad, the cursor stays in place when you lift the stylus; nice straight lines may be created by lifting the pen as you hold the button and placing it down at the desired destination.

When the *Musicland* program scans your work in MUSIC DOODLES, it scans from left to right, turning the lines you have drawn into rectangular notes on the staff. Notes of longer durations are indicated by wider rectangles and may range in pitch from F below the bass clef to A above the treble clef. Sharps and flats are shown by placing the rectangles slightly higher or lower in relation to the lines and spaces; this system is very clear, although it presents a slight challenge in the area between and above the staffs. The program can accommodate a maximum of 120 notes in one screen.

Musicland interprets your drawing as switching on or off elements of a grid 35 units wide and 41 units tall. The grid is scanned from bottom to top, and notes in the same column may not be closer than three half steps. The musical result of this is that a solid line drawn along the side of the screen will become a diminished seventh chord with the lowest element of the line determining the notes of the chord; also, if two notes are placed one or two half steps apart, only the lower note will remain. In the horizontal dimension, a solid line drawn along the top of the screen will become one long note. While transforming shapes you may notice a related effect when two notes of the same pitch are placed next to each other and they fuse into one long note; to create a rhythm on a single pitch you must leave an empty element between each note. If you are attempting to create a set of musical blocks based on a traditional song, the result of all this is that one block may contain two measures of 4/4 time with some possible loss of rhythmic accuracy at the sixteenth note level.

The *Musicland* disk is accessed continuously while the program is being used. Moving from MUSIC BLOCKS to MUSIC DOODLES takes roughly 36 seconds, while moving from there to TIMBRE PAINTING takes about eight. Disk access for individual commands takes roughly three to four seconds per command.

Future Developments and Other Considerations

The Mountain Computer synthesizer cards used by *Musicland* were chosen to obtain optimum musical quality on the Apple and are available from Freelance Musicworks for $295. The University of Toronto Computer Systems Research Group is continuing work which should support this hardware investment. Among the projects currently being explored are a compositional tool which would combine traditional musical notation with the user interface and transformational capabilities of *Musicland* and an intelligent ear training program that would diagnose areas needing improvement and adjust itself to the student.

Musicland is presently being developed for Atari and Commodore computers. Two critical elements in evaluating the success of these adaptations will be the manner in which they utilize the sound chips built into these computers and how the speed of disk access affects the overall performance of *Musicland* on these computers.

The copyright for *Musicland* belongs to The University of Toronto Innovations Foundation, which has licensed the rights for commercial development of the program to Freelance Musicworks. This arrangement allows the Innovations Foundation final approval over the product without acting as a commercial software vendor.

Summary

Musicland is a perfect example of the realization of the idea of a learning environment with no threshold and no ceiling. Its user interface is so transparent and the concepts it presents so powerful that children as young as six can explore musical material normally taught in a college composition class. Unique among computer music programs, *Musicland* places equal emphasis on learning the structure that underlies music and on developing individual creative expression through music. I would consider use of this program in a variety of contexts including elementary, middle-school, high school and college classes and with learners ranging from the musically untrained to the experienced musician.

Publisher's Reply

We are aware of *Musicland*'s limitations: The new version of *Musicland* will have disk storage capability, as well as additional sample compositions to be used in MUSIC BLOCKS and TIMBRE PAINTING. We are also working on making it available for other computers.

There tends to be a popular conception that *Musicland* is just for kids. This is not quite accurate. *Musicland* has in fact been used with college students at the Berklee College of Music in Boston. In addition, one individual won an international composition award for his work with *Musicland.*

John Amarol
Freelance Musicworks

© Copyright 1985 ICCE

BANK STREET MUSICWRITER

By Glen Clancy

Mindscape, Inc., 3444 Dundee Road, Northbrook, IL 60062

Target Age Group: Nine to adult

Subject: Music

Cost: $49.95

Hardware Required: Commodore 64 or Atari (48K), one disk drive, TV or monitor (preferably color), dot matrix printer with graphics capability (optional)

Reviewed by: Jim Ellickson-Brown, Elementary Music Specialist, Mountain View School, Corvallis, OR

Description

Bank Street MusicWriter is based on the idea of creating the musical equivalent of a word processor, where the user can enter and edit three-part (Commodore) or four-part (Atari) musical scores. Music can be played back at any point; each voice can be separately controlled by volume and sound envelope. High resolution graphics provide visual feedback—notes light up as they play. The user may adjust the speed (tempo) at which music plays and obtain printouts of all or individual voices.

Documentation consists of a three-page introduction to the program, a 24-page description of the software features, a 23-page tutorial stepping through entry of an arrangement of "On Top of Old Smokey," an eight-page introduction to music fundamentals, and two pages of advanced techniques. A quick reference card has a flowchart and summarizes program commands; included on the back of the disk are seven sample compositions.

Using the Program

Using *MusicWriter* is easy. From the main menu the function keys are used to highlight one of 12 options; this menu can be reached at any time by pressing the left arrow (or ESC on Atari) key. Selecting NEW PIECE, for example, you are advised that any work in memory will be lost and asked to confirm that you wish to proceed. You are then asked to choose a key signature and taken to the main editing screen, which consists of two grand staffs. The cursor is a small arrow which points to where the next music symbol will appear. Using IJKM (Commodore), the cursor can be moved around the screen, while a prompt at the top of the screen advises you of the pitch name of the cursor's location. Pressing the P key lets you hear a note without entering it. Pressing any of the number keys 1, 2, 4, 8, 6 or 3 enters a whole, half, quarter, eighth, sixteenth or thirty-second note respectively and advances the cursor an appropriate amount. Rests, dotted notes, triplets, sharps, flats and ties are entered by preceding these numbers with other keys. Pressing O makes subsequent notes sound an octave higher than written, the S key changes the direction of note stems, and the V key switches from voice to voice. Pressing the B key enters a bar line—you do not have to enter a bar line at the end of each staff. The shift keys are used to allow quick movement of the cursor keys from side to side as well as switching and scrolling from one staff to another; two of the function keys allow you to go to the first or last staff of your piece.

Additional editing functions include insertion, deletion, turning off selected voices, slurring or tying notes, copying music, erasing notes, repeat signs, and first and second endings. Additional options allow checking for accuracy of time signatures, setting the tempo and altering the sound envelope of each voice. During playback the speed may be increased or decreased using the F and S keys; playback may be ended at any time by hitting the left arrow (or ESC on Atari) key. A variety of disk functions include playing an entire disk (ALBUM) or selected tunes (JUKEBOX), formatting disks, renaming, deleting, copying and chaining pieces (MEDLEY). In addition, the score for any composition may be printed out using a graphics printer.

Suggested Uses

Bank Street MusicWriter has many possible uses. In my classroom I used it successfully with kindergarten through fifth grade students. Three main uses emerged:

1. **As a tool for the teacher to illustrate musical ideas and as an aid in group composition.** When discussing musical ideas with my students, *Bank Street MusicWriter* gave me an additional way to make the symbols of music writing come alive for students. Students were excited by the way the notes lit up as they played and liked the different colors of each voice. The ability to change the tempo makes it easy to adjust the speed to the ability of students to follow the display. One second grade class developed a composition in which

© Copyright 1985 ICCE

two voices travelled from high to low while a third voice jumped by octaves to stay out of the way of the others. The class sang along as they watched the piece unfold; one student who rarely participates actively in class was very attentive and commented on how interesting it was to him. For fourth and fifth grade classes I was able to use the copy function to illustrate the entry of voices in two rounds that the students had studied. When playing the sample excerpt from the Nutcracker Suite which comes with the program, I found it was a good idea to give the students a listening task—they were easily able to find the A-B-A structure of the piece, raising their hands when the contrasting middle section of the piece began and when the first section returned. When using the computer in this way I have found it most effective to have the program loaded and leave the monitor turned off until needed; my class can then engage in their regular music class activities and use the computer as an additional learning tool, rather than having it be the primary focus of the class.

2. **For individual student work in composition.** I quickly found many of my students interested in working with the program during their free time. One fifth grader who takes piano lessons quickly mastered the cursor movement, note entry and copying functions and could work independently; another student was very interested in entering a piece she had performed for the class and dictated for me as she played the tune by heart. A third student wanted very much to work by himself; he soon had notes from all voices scattered over the screen. Playing it back the computer made sense of it, but we found it difficult to follow the score; a valuable discussion ensued. In traditional music notation, notes are written from left to right; notes that sound at the same time are written so that they line up vertically. *Bank Street MusicWriter* follows the first of these conventions but not the second. As a result, music notes sounding simultaneously will light up together but will be lined up in the vertical dimension *only if* you make an effort to do so.

3. ***Bank Street MusicWriter* can be used independently by students as a practice aid,** adjusting the tempo to fit their skill level as they play along with the computer, following the notation. Both my students and I enjoyed this activity. In using the program in this manner, I found it was a good idea to include an introductory section of steady beats during which the user can adjust the tempo and a contrasting four-beat phrase signalling that the piece is about to begin.

Strengths

Bank Street MusicWriter excels in its graphics. Scrolling from one screen to another is very fast and produces the effect of a turned page rather than scrolling. This is accomplished technically by beginning the scrolling while the last notes of a staff are being played. The highlighting of each voice as it plays is also very effective in helping students follow the notation. The main menu is easily accessible from anywhere in the program, prompts are very clear, and incorrect key presses do not lead to disaster. Disk access is reasonably fast and users are able to select files using a convenient menu format.

"BANK STREET MUSICWRITER excels in its graphics. Students were excited by the way the notes lit up as they played, and liked the different colors of each voice."

Weaknesses and Quirks

As excited as I am about *Bank Street MusicWriter,* I found several areas for possible improvement:

1. The tempo selection provides 16 possible tempos. Five of these are in the range of 72 beats to 224 beats per minute, while the remaining 11 are in the range of 26 to 64 beats per minute. It would be much better to have a larger range of moderate to fast tempos to choose from.

2. The copy function lets you make one copy of a passage. Music frequently consists of one musical idea used repeatedly. It would have been very exciting to be able to take one idea and copy it repeatedly; as is, one has to repeatedly cut out the part to be copied. It would also be nice to have an UNDO function in case a mistake is made in placement; this is especially irksome since the DELETE function works only with all voices.

3. The tutorial is excellent in its thorough description of entering notes. The harmonization used, although rhythmically quite clever, appears to my ear to be in the wrong key (C would have been more appropriate than F).

4. The raspberry sound which greets an incorrect key press is a little louder than necessary. I would prefer an option of no response and silence.

5. *Bank Street MusicWriter* appears to be designed for the home market. Teachers using the program must rely on their own skills to develop a method for teaching the program to their students. A more comprehensive disk library of compositions would be a nice addition.

And, among the quirks:

1. The keys to select octave higher and the natural sign lie very close to the cursor control keys. It is very easy to select these accidentally.

2. When you delete a note, the cursor advances. To stay in the same place you must press the space bar, then the J key to back up.

3. It takes some time to get used to the idea of the cursor pointing *to* rather than being *at* the point where a symbol will be entered. When copying music it is important to note the vertical position of the cursor when you mark the beginning of the passage to be copied; otherwise, you may wind up with an undesired transposition.

Summary

Bank Street MusicWriter is of use to both the novice and experienced musician. The software performs very well with exceptional graphics. The documentation is well done with clever dialogue and illustrations; the user interface is consistent and easy to understand. *Bank Street MusicWriter* is not and was not designed to be the most powerful computer music editor, but it can provide you and your students with a valuable tool for learning, exploration and self-expression.

Publisher's Reply

Mindscape appreciates Mr. Ellickson-Brown's careful review of *Bank Street MusicWriter* by Glenn Clancy. We would like to reply to a few of his points of criticism.

MusicWriter was specifically designed to be as powerful and easy to use as possible. The copy function works accordingly, in that the size of the passage to be copied is not limited to the few measures one is usually allowed in programs offering repeat copy and UNDO options. With the power offered in *MusicWriter,* whole sections of music can be reproduced with one action. The requirement to use markers and follow

© Copyright 1985 ICCE

on-screen prompts with each instance of copying may seem repetitious but it promotes ease of use by eliminating any possibility of the user getting lost. Last, the review mentions the need to place the cursor vertically when copying to avoid "undesired transpositions." We feel the ability to easily transpose while copying, simply by placing the cursor at a new starting pitch, is one of *MusicWriter*'s best features.

Ease of use was also considered in choosing keys to select Octave and the Natural sign. The "O" for Octave and "N" for Natural are easy to memorize and understand. The proximity of these keys to the cursor control keys results from keyboard layout on the Commodore only.

Once more, ease of use is exhibited by the fact that the cursor advances when a note is deleted, allowing the user to delete a whole line of notes without advancing the cursor each time.

Finally, the review states that *MusicWriter* was not designed to be the "most powerful music editor." Glen Clancy's goal was, in fact, to design the most powerful music editor available for microcomputers. Judging from a thorough analysis of other products, we feel this goal was achieved: *MusicWriter*'s word-processing approach to composing and editing; full range of notes and rests; ties, triplets, endings, repeats and other musical markers; ability to easily copy, delete, transpose, check time signatures and print out scores; plus many other features make *MusicWriter* a very powerful composing and editing tool.

Scholastic, Inc.

© Copyright 1985 ICCE

IV. Student Activities

In this section, we are told to expect activities and reproducible worksheets which are adaptable for elementary, middle and high school students, and which, when combined with the software, form the "basis of individual science lessons, complete units on biology or anatomy, or special curriculum modules that cut across several subject areas."

Specifically, this section includes:

- A Laboratory Record Sheet for use during dissection and reconstruction.
- Eight student worksheets which correlate with the information provided through the magnifier icon in the software. The diagrams on these pages are awful.
- Additional worksheets concentrating on the heart, animal classification and location of organs.
- Twelve pages of suggestions for related activities in science, art, language arts and social studies. These suggestions were not particularly inspiring, but were nonetheless appreciated by teachers.

Using the Program

Manual instructions for starting up the program are uncomplicated and accurate.

The Main Menu offers four choices: Perform a Dissection, Reconstruct the Frog (Beginner Level), Reconstruct the Frog (Expert Level) or Quit.

The "Dissection Screen" consists of five elements:

- A ventral view of a frog with skin and abdominal muscles removed. Two more layers of organs are exposed in turn as the dissection progresses. The organs are colored. (We tried operating the program using a monochrome monitor, but the organs were very difficult to distinguish.
- A "tray" holding scissors, forceps, probe and magnifier.
- A ruler marked in English and metric units.
- An "examination area" which shows shaded outlines of each piece of tissue. There are 23 outlines.

The screen also shows a movable hand which is used to "point" at the desired tool, organ or receiving outline. The hand can be steered using the keyboard, joystick or Mouse. We tried all four and found the Mouse most responsive, followed by a self-centering joystick. The keyboard option requires use of the IJKM Keys or, on the Apple IIe, the arrow or IJKM keys. One thoughtful feature which the developers included won friends quickly. When it is controlled by a joystick, the pointer takes increasingly bigger "steps," so the further it moves the faster it moves.

Each organ is removed in its turn by:

- Using the probe to locate its snipping point.
- Snipping the organ with the scissors.
- Carrying the organ to the examination tray with the forceps. (Many students became bored. They felt these steps took too long.)
- Examining the organ with the magnifier.

The examination reveals screens full of textual information and diagrams. Some of the diagrams are animated, which students like. The animation of the circulatory system was too fast, however. There is no way to slow down the motion, nor to make it pause.

Several students remarked on the availability of the information. They preferred this method over "looking for it in a lot of books."

If the incorrect icon is selected, or if an attempt is made to use it inappropriately, it simply refuses to react.

If "H" is pressed at any time, a window opens to display a suggestion about what to do next.

Soft clicks and beeps (which can't be turned off) signify correct instrument selection, successful snipping, etc. Often, however, the sounds are too quiet to hear.

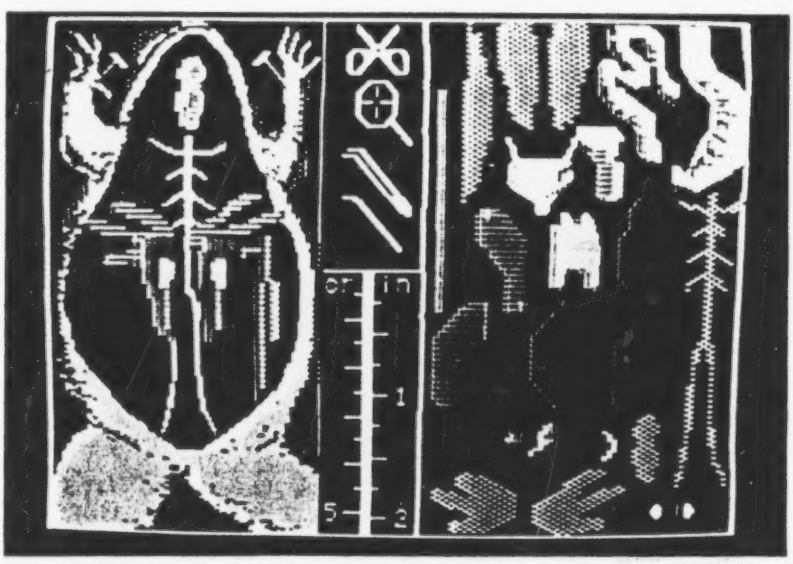
I watched students snip and remove organs in the "wrong" order, i.e., not according to the advice given by the "Help" window. As long as they didn't attempt the impossible, such as removing an organ before removing the organs above it, the instruments did their job as usual. The instruments would not function on organs which were not "available."

Reconstruction

The dissection does not have to be completed before reconstruction begins. Organs may be removed and replaced at will. The capability to rebuild the frog offers a unique opportunity for reinforcement of learning.

Reconstruction of the frog can proceed at a beginner or expert level. Beginner level reconstruction is simply a dissection in reverse. Organs can only be replaced in their correct sequence. The "Help" function identifies errors and suggests which organ to replace next. The magnifier functions as it did in the dissection, revealing the same information and diagrams as before. Some of the diagrams are helpful for deciding where to place organs. In actual fact, the program does not allow organs to be replaced incorrectly.

At the end of a reconstruction, or after the student presses "Q," the computer responds with a score (maximum



100) which rates the students' performance.

Reconstruction in the expert mode is identical with two exceptions. Points are lost if an attempt is made to replace an organ incorrectly. In addition, points are lost each time the student requests help.

Summary

While *Operation: Frog* should not replace an actual dissection, it would be a worthwhile component of a grade 7-12 science program. It offers an opportunity to practice the procedure several times without some of the aggravations and confusion of real tissue dissection. Errors are handled routinely and the program includes readily accessible hints. The price paid for these conveniences and enhancements is, of course, a loss of realism. No odor of formlin, no bones, texture and perspective are absent, and the organs are colored unrealistically for easy identification. And the frog is male, although information about the female urogenital system is included.

Yet, we have no reservations recommending *Operation: Frog*. It's a winner.

Publisher's Reply

As the reviewer states, *Operation: Frog* indeed should not replace an actual dissection, but should be considered as an effective learning activity in its own right.

Second, in this program more than most, we (Interactive Picture Systems, the software developer, and Scholastic) found ourselves stretching the limits of microcomputer technology, requiring us to make some compromises. For example, the restricted capacity of floppy disks meant reducing the number and scope of animation sequences, and the limited number of screen colors forced us to use "unnatural" colors to distinguish some organs. In almost all respects, however, the procedures, graphics and concepts in *Operation: Frog* remain true to the actual dissection experience. Our science advisors made sure of that.

Although the different patterns that distinguish the organs make it possible to use the program on a monochrome monitor, we agree that it is better in color. And we also agree with your reviewer—it is, indeed, "a winner."

Philip Miller
Editorial Manager, Scholastic Software

TESS (The Educational Software Selector) 1985 Edition
By Bob Haven

EPIE Institute, Box 839, Water Mill, NY 11976; ph. 516/283-4922
1985, 785 pages, $59.95
ISBN #0-916087-00-X
Reviewed by: Dick Ricketts, ICCE

Compared to *TESS* '84, the word for *TESS* '85 is **more. More** coverage—an estimated 80 percent of all commercially available education programs available in the United States; **more** money—$59.95 vs. $49.95; **more** unique entries—6,400 vs. 4,200; and **more** value for the money (proof is left to the reader). Regan and Doris Carey's advice in the May 1984 *TCT* applies even more to *TESS* '85:

> ***"TESS may be expensive for the individual user, but it is a must for resource centers, libraries, school district offices, and any group responsible for ordering educational software. TESS is an investment that cannot be overlooked."***

The usefulness of *TESS* can be tested by example. Suppose I'm looking for a good readability program. Not being sure what category I'd look under, I skip the table of contents and find "Readability Analysis, 53" in the subject index. Page 53 lists 10 programs. Only one entry has a review listed; neither EPIE nor MicroSIFT has evaluated any. Three programs came out in 1984. The first publisher-furnished description doesn't specify formulas available or editing capabilities, the second lists the seven most common readability formulas but doesn't mention editing, and the third is:

> ††8) **READABILITY ANALYSIS**
> **Cat.** 105 **Rel.** 03/84
> **Types:** Teacher aid, comput'l.
> **Grades:** 1-12
> **Uses:** School; for main curric., remedial, special ed.
> **Grouping:** Indiv., small group, class.
> **Description:** Teacher types in appropriate passage and the computer counts syllables, words, and sentences and calculates average sentence length and word difficulty. Reports these data as well as reading level of text. The teacher may choose from three readability tests: Spache, Dale-Chall, and Fry. Text is easy to enter and two opportunities for editing are provided. Average lesson is 10 minutes. Is copy protected, is list protected. Network version not available.
> **Configuration/Price:**
> 1. TRS-80 mod III/IV, 48K, 5" dsk dr., Disk BASIC, TRSDOS; on disk $26.95.
> 2. Comm. 64, 64K, 5" dsk dr.; on disk $29.95.
> 3. Apple II+/IIe/Compatibles, 48K, 5" dsk dr., Applesoft, DOS 3.3; on disk $26.95.
> **Components:** Diskette, teaching guide.
> **Availability:** Gamco Industries, Inc.

The Software Descriptions section explains that †† marks programs appearing in *TESS* for the first time. The Software Suppliers section provides me with Gamco's address and phone numbers. I discover that Gamco negotiates quantity discounts, offers a money-back guarantee for previews and returns, makes "back-up copies available at $15 at time of purchase," and provides any updates upon request.

The cost of *Readability Analysis* is less than most other programs listed, Spache and Dale-Chall are respected formulas, and I've always liked the Fry. Perhaps *Readability Analysis* will meet my needs. However, the section detailing "Seven Steps for Responsible Software Selection" reminds me that *TESS* should help during screening, not dictate purchasing decisions.

TESS also may be useful in an unexpected way—the absence of some programs in a particular subject area may call attention to a needed area of development. For instance, no viable simulations appear in the Driver Education list. Developers, educators and students can look through *TESS* for gaps and thus concentrate on creating programs that may satisfy unmet needs.

Incidentally, a fairly intensive search revealed that the most impressive aspect of *TESS* is something humbler than accessible layout, comprehensiveness, and to-the-point information. No typos or inaccuracies were uncovered.

Users of *TESS* will, however, be annoyed by two intractable difficulties: Relatively few evaluations and instant datedness. Although 1,500 entries include evaluation or review citations, that is less than 25 percent of the total entries. This problem will remain unsolved until evaluation/review organizations cooperate more closely, the feedback cycle is speeded up, and more money is forthcoming to support their work. The datedness occurs because it takes almost a half year after the last information is entered to publish *TESS*. This problem will be ameliorated when *TESS* goes on CompuServe with bimonthly updates.

Despite the value of *TESS*, its price of $59.95 attracts attention. EPIE is willing to discuss quantity discounts, and groups may find it worthwhile to investigate. **END**■

© Copyright 1985 ICCE

Software REVIEWS

Linda Rathje, Editor

READER RABBIT AND THE FABULOUS WORD FACTORY

The Learning Company, 545 Middlefield Road, Menlo Park, CA 94025

Grade/Target Age Group: Ages five to seven

Subject: Reading readiness

Hardware Required: Apple II and II+ (64K minimum), Apple IIe, IIc, Commodore 64, IBM PC (128K), IBM PCjr (128K), IBM PC XT, IBM PC AT, IBM PC Portable, Atari (48K); 1 disk drive; monitor or TV (color recommended)

Optional Hardware: Joystick, KoalaPad, Muppet Learning Keys

Cost: $40

Reviewed by: Scott Turner, *First Grade Teacher, Washington Elementary School, Vancouver, WA*

- Reader Rabbit
- The Puzzler
- The Newsroom
- Cardiovascular Fitness Lab
- Fay's Word Rally

Reader Rabbit is a fun, easy-to-use program that enables children to practice reading and prereading skills using 206 phonetic consonant-vowel-consonant (CVC) words and 72 pictures. For the teacher, *Reader Rabbit* motivates students to eagerly practice these essential skills, but unfortunately provides no record of results.

A clearly designed menu offers a choice of four games. To control operation during the games, the user may choose the keyboard (arrow keys to move and space bar to take action), joystick, KoalaPad or Muppet Learning Keys.

1. SORTER

In the first game, Sorter, the student is shown or selects a target letter and position (initial, medial or ending). Words are then presented one at a time and the child must decide whether or not each matches the target format.

2. LABELER

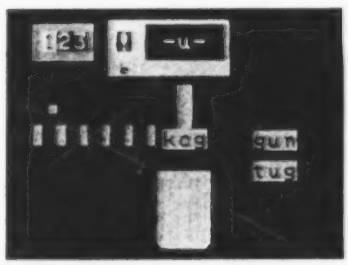
In this game the child must build CVC words to label pictures. Letters are picked from groups of three beginning and ending consonants, and medial vowels.

3. WORD TRAIN

In this game, the child builds a word train. A three-letter word in the engine is the first word in the train. Three new word choices are then displayed, and the one chosen must differ from the previous train car by only one letter. This is the hardest game for most children to master. It forces them to notice the letters and their positions within the words.

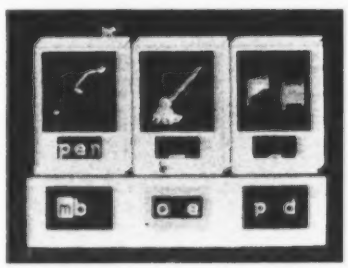
4. MATCHUP

Matchup is a concentration game using 12 cards. The backs of the cards are blank and are chosen by moving a "hand" to the choice and then pressing the space bar. Besides the standard picture-to-picture matching, the menu also offers picture-to-letter, picture-to-word and word-to-word matching choices.



SORTER

- Identifying unfamiliar letters.
- Gaining speed in recognizing letters.
- Locating beginning, middle and ending letters in C-V-C* words.



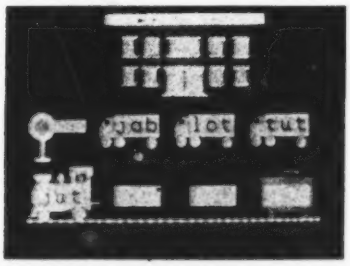
LABELER

- Locating letters in their beginning, middle and ending locations in C-V-C words.
- Noticing middle vowel location in C-V-C words.
- Developing spelling skills.

Reader Rabbit provides an excellent way to practice phonetic reading skills. The displays, animation and sound are all appropriate for primary-age children and do not disrupt the flow of the program or the rest of the classroom. Motivation to use the program remained quite high through many months of use in my first grade class.

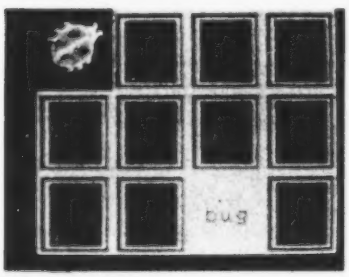
Two of the four games allow for some prescriptive modifications to meet individual needs. Sorter lets you regulate the speed and choose specific letters and positions to be worked on. Matchup lets you choose what type of matching is to be done. Pictures available with Labeler and Matchup can be specified by group,

© Copyright 1985 ICCE



WORD TRAIN

- Identifying words that differ by one letter.
- Using logic to sequence words: holding two variables (letters) constant while a third changes.



MATCHUP

- Associating and matching pictures with pictures, words and parts of words.
- Matching words with words.
- Improving spatial awareness (up/down, left/right).

*consonant-vowel-consonant

but the groups are created by subject area (e.g., animals, people, house, containers, travel) rather than by phonetic type.

The lack of a management feature was disheartening and is a major omission. The only way to judge a child's grasp of the program skills is to watch the screen or listen for the auditory feedback. Teaching responsibilities make these options haphazard, if not impossible.

CONCLUSION

Reader Rabbit is a highly motivational game that has educational validity for any reading teacher. The program is well designed and easily used by young children. A record of student results would make it much easier to manage in the classroom, but in the final analysis I just can't knock success. My kids absolutely love this program. It is one they ask for by name.

Publisher's Reply: None.

THE PUZZLER

Sunburst Communications, Inc., Pleasantville, NY 10570

Grade/Target Age Group: Grades three to six

Subject: Reading/Thinking Skills

Hardware Required: Apple II series (48K RAM), 1 disk drive, monitor

Cost: $59.00

Reviewed by: Mary Ellen Sanders and Kathy Heim, *Special Reading Teachers, Erie Elementary Schools, Erie, IL*

The Puzzler is designed to foster predicting and confirming strategies in reading. Teachers familiar with Russell Stauffer's directed reading-thinking activity (DR-TA) will be immediately at home with this program. This package consists of two short introductory story activities for the teacher to use with a large group via the overhead projector, and five stories on disk. During the introduction students read a short story in sections and, drawing on clues in the content, make predictions about who or what something is in the story. As more of the story is read, students are asked to confirm or reject their predictions, or make new predictions. The end of the story does not reveal one "right" answer. After completing the introductory activities, students can work individually, in pairs or in small groups to read the five stories on disk, record their predictions at the computer as they read, and compare their responses to those of other students who read the stories during field testing. The teacher then leads a discussion in which students discuss their predictions and what prompted them.

Strengths

Reading is presented as a thinking process in this program, encouraging readers to use higher-level thinking skills and to become actively involved with the reading process.

The format of *The Puzzler* is very motivational. Students use their own background knowledge, watch for clues, and read between the lines to make their own predictions. And since there is no one right answer, they are not discouraged from making responses different from their friends'. In fact, the program can't proceed until predictions are read and refined.

The documentation is well organized and very easy to use and it supplements the program nicely. Actual copies of the stories are included. The whole-group introductory activities on prediction, titled "Petoskeys" and "The Blog," encourage students to make many predictions which fit all the clues provided. When the computer stories are introduced the students are already comfortable with the format; self confidence has been established and leads to much positive interaction between readers.

The Puzzler is not only a good piece of software, but Sunburst offers a very generous lifetime guarantee; just send the damaged disk back for a free replacement. And there's a toll-free number to call with any minor problems.

Petoskeys

The boy was looking for petoskeys.

He was walking slowly to make sure he wouldn't miss them.

Each time he looked, he found a number of them.

Petoskeys are not easy to find because they are almost the same color as the sand.

Weaknesses

Compared to the strengths of this program, the weaknesses are minor. We found the mechanics of entering predictions to be confusing initially.

We have some concerns with two of the five stories. One story, "What's Missing," is more frustrating than the others as the clues are not sufficient to satisfactorily narrow the predictions. Another story, "The Dangerous Stuff," has the opposite problem; from early in the story the clues point to one solution, so the readers are not motivated to further refine their predictions. This could, however, be a good way to introduce the disk.

© Copyright 1985 ICCE

The lack of any record-keeping utility on the disk or in the documentation is a drawback when discussion takes place later. The documentation suggests students take notes concerning their predictions to refer to during discussion, but a printout, or at least a record-keeping sheet, of each student's predictions and at what point in the story they were made would be helpful. However, this is not a problem if discussion takes place as the story is being read.

"The objective of fostering 'the ability to use predicting/confirming strategies in reading' can be met without the use of a computer through existing teaching strategies . . . but THE PUZZLER complements the teaching of such strategies very nicely."

Suggested Uses

The documentation indicates *The Puzzler* was originally designed for students in grades four through six, but it is being used with students in grade three through adult education. As elementary special reading teachers, we work with small groups of three to six students whose reading achievement ranges from two years below grade level to above grade level. We have found the program to be effective with students in grades three through five. We have used the introductory lessons in a self-contained third grade classroom, and we have used the program by reading to second graders who orally make their predictions for us to enter at the computer. We find the program very usable in our setting. It would also work well with a classroom reading group, preferably with no more than six students as there is space for only six predictions at one time in the story.

Recommendation

The Puzzler is motivational, has well-organized, practical documentation, is inexpensive, and has a lifetime guarantee. The objective of fostering "the ability to use predicting/confirming strategies in reading" can be met without the use of a computer through existing teaching strategies such as the DR-TA. But *The Puzzler* complements the teaching of such strategies very nicely. Our students enjoy using the program, and we recommend it for classrooms or special classes where educators are interested in developing higher-level thinking skills and teach reading as a thinking process involving an interaction between the reader and the print.

Reference

Stauffer, Russell G. "Chapter 2—Group-Directed Reading-Thinking Activities." *Directing Reading Maturity as a Cognitive Process.* New York: Harper & Row, 1969.

Publisher's Reply: None.

Copyright © ICCE 1986

Software REVIEWS

Linda Rathje, Editor

THE OTHER SIDE

Tom Snyder Productions, Inc., 123 Mt. Auburn St., Cambridge, MA 02138

Grade/Target Age Group: Ages 12 to Adult

Subject: Social Studies—global conflict simulation

Hardware Required: Apple II+, IIe, IIc (48K RAM); 1 disk drive; modem optional; a 30-foot cable for connecting two computers is available for $15.

Cost: $69.95

Reviewed by: *Regan Carey, Ph.D., Doris Carey, Ph.D., University of Nevada, Las Vegas*

- **The Other Side**
- **Three Math Worksheet Generator Programs**
  - **MATHEMATICS WORKSHEET GENERATOR**
  - **E.A. MATHEMATICS WORKSHEET GENERATOR**
  - **MATHSHEET**
- **ThinkTank 512**

"Do you think we should send them more money?"

"How do we know they'll use it to build the bridge?"

"How do we know they won't?"

"It seems to me that we've helped them more than they've helped us."

"That's true. But we need them to. . . ."

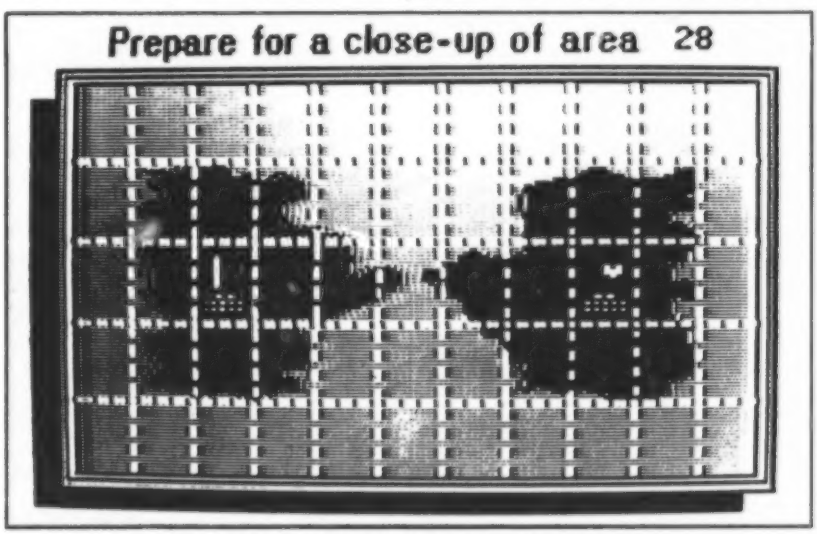
. . . And so goes a typical discussion between members of one team playing *The Other Side*, a game which, according to its publisher, "simulates the fragile balance between two nations . . . (and) can only be won through negotiation and cooperation."

Two features of the package caught our attention before we ever booted up the software. One was the collaborative environment which, given Tom Snyder's excellent track record with the *Search* series and *Snooper Troops*, we felt would result in an educational simulation that could not be imitated by board games or other non-computer media. The other was the 30-foot cable for linking two microcomputers (in our case, Apple IIe's)—the promise of true interaction finally fulfilled. We were not disappointed.

Program Description

The Other Side is a game for six or more players, with three players to a team. Though the optimum setting for the game is two microcomputers in different locations, it is possible to locate both teams in one room as long as each team does not overhear the other team's decisions. Furthermore, the game can accommodate play on a single computer, with teams taking turns at the computer and using time between moves to plan strategies for their next move. The game includes built-in communications routines so it can handle play between distant teams. Full instructions for use with Hayes Micromodem and Apple Modem are provided. Play may be easily interrupted and resumed later, since all data is automatically saved as play progresses. We consider this feature particularly attractive in a school setting where a game may span several class periods.

The object of the game is to achieve communication and collaboration between two nations. The world represented on the game boards consists of two island nations with a narrow strait between them. There is a focus on international collaboration: The two nations must cooperate to build a bridge over the strait. Two different games can be played. The first is a collaborative game

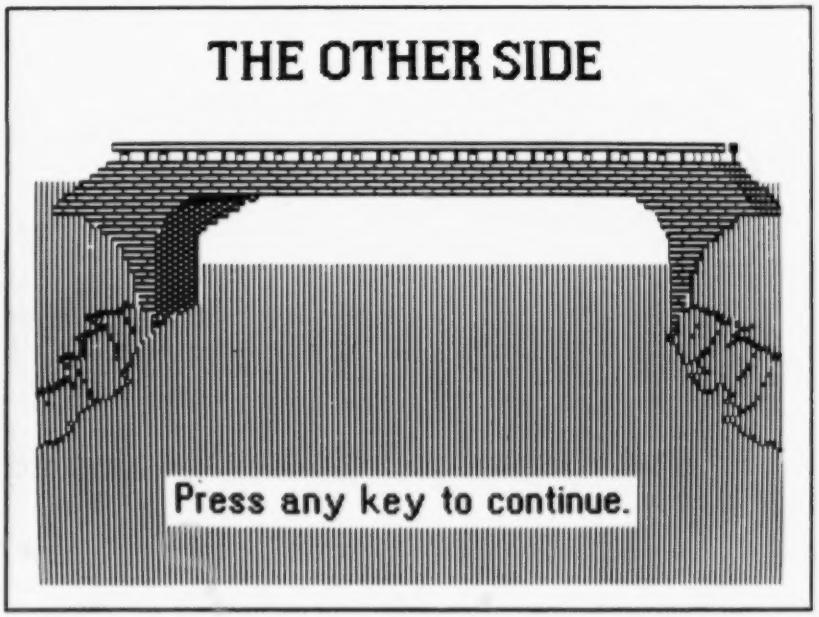


The two countries of ***The Other Side*** *explore the world for natural resources.*

© Copyright 1985 ICCE

in which the goal is to complete construction of the bridge in as few years as possible. The other is a competitive game in which the side that adds the last brick to finish the bridge wins the game.

An immediate concern of educators who are considering simulations for the classroom is their level of complexity. It has been our experience that historically accurate simulations take into account an enormous number of variables that prohibit their use for enhancing existing curricula. But simpler simulations fail to convey the difficulty of international communication, even when countries aspire to cooperate to achieve peace. How often have we, as educators, heard our students preface their solutions to world problems with, "All you have to do is—"? *The Other Side* succeeds in achieving the delicate balance between complexity and playability by presenting factors that affect the outcomes of international negotiation and domestic concerns. These include each country's resources (represented by fuel in this game), economy and trade (mixing and selling fuel), and technology (bombs).



The goal of the game is to build a bridge between the two countries. Each brick added to the bridge costs progressively more money.

Each country can accumulate cash through fuel exploration and drilling, including prospecting forays on "the other side." They can drill for three types of fuel. Each side has a unique type of fuel (orange and green, representing each country's color on the game board), and a third type, blue, is available to both countries. Mixtures of all three types are the most valuable. Foreign trade is simulated by exchanges of fuel that benefit both countries, and it is clear that trade must occur for most profitable mixing. Of course, it is possible to resort to aggressive action to take over the other country's fuel.

With cash, each country can buy bricks with which to build the bridge. At the outset, a brick costs a mere ten dollars, but the price increases with each subsequent brick. You can imagine the cost of the last brick if one country has to do all the work! Realistically, though, the game makes this very difficult by the inclusion of the villainous CAD (an unfortunate duplication of the acronym for Computer Aided Design)—which stands for Computer Aided Defense. In a nutshell, CAD intervenes "quickly and dramatically" to take over when the country's economy or well-being are endangered by the players. CAD is not at all interested in the bridge or world peace, and overriding CAD is expensive. This feature effectively conveys the message that decision makers must work for a careful balance between domestic harmony and international cooperation.

The author needed a symbol for technology, realizing that the instrument devised should have the dual purpose of benefitting fuel exploration in the home country and providing aggressive or defensive action against the other side. A bomb was selected. We thought the bomb an unfortunate choice, suggesting an aggressive and negative use of explosive devices. We can sympathize with the author's reluctance to avoid the use of nuclear energy, though, since this is not a simulation of nuclear war and is probably not intended to specifically ride the popular wave of disarmament discussion.

One of the best features of the game is its level of difficulty. It is very difficult to "win." As teachers, we were comfortable with the game's objectives. We felt that having students observe the complexities of international cooperation was the most important educational objective achieved with this simulation. An added benefit is the necessity of defining roles for the players. It is virtually impossible for a single student to dominate the game and record all of the in-

© Copyright 1985 ICCE

formation that scrolls by in different sections of the screen at any given time. In other words, team cooperation and inter-team communication are the purposes of the game.

Players are limited to communicating only through the computer "hot-line," with each message limited to 30 characters while the clock is ticking away. Trying to guess the other side's intentions and questioning one's own assumptions about their actions are central to the strategy of the game. They are also the most obvious points of discussion in a debriefing session.

We cannot overemphasize the role of the teacher in this simulation. Though it is difficult to interrupt the two-computer game and discuss events as they occur in real time, the real benefit of the game is bound to be in the ensuing discussion about the players' decisions and their outcomes.

Though the game itself is captivating and would do well as a stand-alone parlor game, it is a powerful simulation for classroom use. The link with world peace, politics, international negotiation, trade and economics is not obvious in the game and must be made explicit by the teacher so students understand the game's purpose is to initiate discussion about international communication and cooperation. By the same token, the most important features of the game can be missed if the game is used without a teacher's follow-up.

Recommendation

The Other Side is not just a well-designed and creative example of CAI; it could be the beginning of a new chapter in instructional simulation design. We are enthusiastic about its use in the social studies classroom and the range of its potential for teaching advanced concepts in communication, economics, political science and group dynamics. Educators have long awaited inexpensive systems that provide some of the benefits of mainframe or networked systems. The reasonable cost of this program, combined with the author's sensitivity to the need for activities that take into account time and equipment constraints in our schools, have led us to recommend *The Other Side* for use in social studies classes. We have been very pleased with its potential for the discussion of teaching strategies in teacher training settings, and look forward to introducing it as the best example of a manageable global conflict simulation that we have seen to date.

Computer Literacy for Elementary and Middle School Students

- 12 GREAT INTRODUCTORY ACTIVITIES
- DESIGNED FOR ELEMENTARY AND MIDDLE SCHOOL USE
- HANDS-ON BOOKLET CONTAINS NECESSARY BACKGROUND INFORMATION
- EASILY INTEGRATED INTO CURRICULUM
- MOST ACTIVITIES DO NOT REQUIRE COMPUTER ACCESS

8½ x 11 • 56 pgs. • ISBN 0-924667-02-8

Published by the International Council for Computers in Education, the leading U.S. and international professional organization for computer educators. To order the above booklet or for more information about ICCE membership and publications, write or call:

ICCE
University of Oregon • 503/686-4414
1787 Agate St., Eugene, OR 97403 USA

© Copyright 1985 ICCE

THINKTANK 512

Living Videotext, Inc., 2432 Charleston Road, Mountain View, CA 94032

Grade/Target Age Group: Ages 12-adult

Subject: Planning, writing and studying

Hardware Required: Macintosh 512K, ImageWriter printer; external disk drive optional

Cost: $245

Reviewed by: *Lynne Anderson-Inman, Ph.D., College of Education, University of Oregon, Eugene, Oregon*

ThinkTank is not a word processor, it's an idea processor. Designed to assist users in the task of organizing their ideas, it operates on the principle that thoughts arranged into an outline are far more useful to us than thoughts which are not arranged at all. *ThinkTank* has been created to assist in this arranging process, but not to direct it.

ThinkTank 512 is the latest entry into the innovative *ThinkTank* line. Although all programs carry the *ThinkTank* name, some of their features vary and so does their utility to teachers and students. This review is specifically on the *ThinkTank 512*, a program designed to take advantage of the intuitive environment and extended memory of Apple's 512K Macintosh. As such, it is the most versatile and the most sophisticated of the *ThinkTank* programs I've tried. After using *ThinkTank* on a "Fat Mac," it'll be hard to settle for anything less.

How It Works

Constructing an outline is as easy as making a list. Ideas are entered as words, phrases or even sentences. Pressing return between ideas allows each "headline" to appear as a separate line of text. If an idea seems to be subordinate to a related idea, a simple click and drag with the mouse moves it to the right. Entries can be arranged and rearranged (up, down, left, right) with movements of similar ease. Users may enter their ideas as random thoughts to organize later, or enter them by level, gradually adding supportive detail in layers of increasing depth. When a heading is moved to another location in the outline, all its subheadings move with it.

So much for constructing an outline.

© Copyright 1985 ICCE

Yes, it's easier and faster than doing the same thing on a word processor, but if that were all the program offered it wouldn't be worth the money. Once an outline is constructed, or even during its construction, the information stored and organized using *ThinkTank* can be viewed and used in some interesting and educationally relevant ways. Headings and subheadings are all preceded by either a plus sign (+), indicating that the heading has a subheading, or minus sign (−), indicating no current subordinate levels. With a double click on any heading preceded by a "+," all subordinate entries can be made to appear or disappear. This allows the user to view the outline in varying levels of detail: Scan just the major ideas, zoom in for the specifics in any selected section, or expose the entire outline for study.

A special feature of the *ThinkTank 512* is its capacity to handle paragraphs of text in addition to the words and phrases that comprise the outline. Double clicking on the headline's leader character ("+" or "−") makes a window for text editing appear. Using a well-designed built-in word processor, users can insert anything from short descriptors to more than 16 pages of text under any idea. If headings and subheadings are rearranged, the text accompanying that idea automatically goes with it. A similar feature is available for graphics. Any image that can be placed on the Macintosh Scrapbook can be imported into a *ThinkTank* outline. To view just graphics in an outline, the user can select another special feature called "Slideshow."

ThinkTank outlines are not limited to on-screen use. A variety of printing options allows the user to obtain a permanent copy of any outline created. If desired, a numbering system can be added to the printed outline, and headlines at varying levels can appear in bold type. The program also facilitates automatic printing a table of contents for any outline, a useful feature when outlines grow to several pages. When only a portion of the outline is to be printed, the user can indicate this by merely clicking the headline of the desired section.

Applications for Students

One of the most obvious educational applications for *ThinkTank* is as an aid to creative writing. For example, students can use the program during the prewriting process to brainstorm possible topics for a composition. The program's easy outlining capabilities can then be used to record and arrange possible subtopics for some of the most interesting ideas, eventually leading to the selection of the one topic that elicits the greatest number of subtopics or the most interesting array of subtopics. Used in this way, the program can help novice writers explore what they know and make decisions about composition content.

For writers who have chosen their topics, *ThinkTank* is an excellent tool for organizing and gradually expanding the information to be shared. Figure 1 is the printed version of the outline created for writing this review. I started by listing the ideas which now appear in bold print. The subtopics were added as I moved back and forth, asking myself what I wanted to say under each heading. For some of the ideas, whole paragraphs seemed to leap into my mind and I used the text processing windows to add these, and then hid them from view as I continued work on the actual outline. While writing I changed my mind about the order of a couple of

1: What it is

2: How it works
- Constructing and using outlines
- Adding text
- Adding pictures
- Printing

3: Applications for students
- Creative writing
- Organizing information for reports
- Study guides

4: Applications for teachers
- Lecture outlines
- Unit planning
- Anecdotal records on students

5: Problems and Limitations
- Better info in manual on printing options
- Teacher's guide and student manual for classroom use
- Samples of educational applications
- Integration with word processor for final product

6: Recommendation
- Positive summary
- Future prediction

Figure 1.

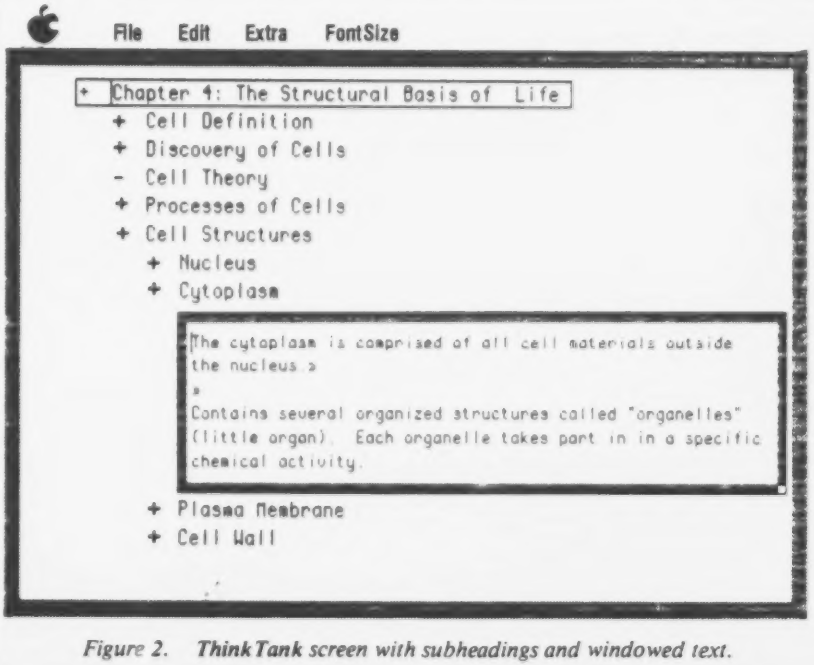

File Edit Extra FontSize
+ Chapter 4: The Structural Basis of Life
+ Cell Definition
+ Discovery of Cells
- Cell Theory
+ Processes of Cells
+ Cell Structures
+ Nucleus
+ Cytoplasm
The cytoplasm is comprised of all cell materials outside the nucleus.
Contains several organized structures called "organelles" (little organ). Each organelle takes part in in a specific chemical activity.
+ Plasma Membrane
+ Cell Wall

Figure 2. ThinkTank screen with subheadings and windowed text.

© Copyright 1985 ICCE

ideas, but the outline was quickly rearranged and reprinted to reflect my revised thinking.

ThinkTank 512 has some features that make it particularly useful to students for planning and organizing longer papers or reports. Because an outline is almost infinitely expandable (up to 3200 headlines and more than 16 pages of text per headline), the program facilitates the collection and arrangement of information from a variety of sources. In addition, a search feature allows the user to find any given word (or part of a word) anywhere in an outline. This has the effect of turning the outline (and its accompanying windows of text) into a data base, allowing students to access and rearrange information through the use of keywords.

ThinkTank 512 can also be used to create electronic study guides. Important information from a textbook can be entered to create an outline of a chapter or unit. Critical details, illustrative examples, definitions of key concepts, and even related charts and graphs can be inserted using the text and graphics windows. Supportive information from lectures or supplementary readings can be added where appropriate. When it comes time to study for a test, the student can manipulate the outline (i.e., expand and contract its headlines) to review the information contained and explore various relationships. The ability to hide and then reveal subheadings or windowed text facilitates a form of self-testing which is extremely effective as a study technique but difficult to execute on paper. Figure 2 illustrates this application with a section from a biology chapter on cells. A printed version of the study guide could be used for review during those times when the computer isn't available (see figure 3).

Nucleus
Definition
The nucleus is a large oval or spherical body near the center of the cell.

The nucleus is the control center for all cell activity.

Components
Nuclear membrane
separates the nuclear material from other parts of the cell, but is not a solid barrier. This allows substances to pass from the nucleus to the rest of the cell.

Nucleoplasm
contains fine strands of chromatin material—thought to be the active forms of chromosomes, which control the activities of the cell

Nucleolus
takes part in protein synthesis

Cytoplasm
The cytoplasm is comprised of all cell materials outside the nucleus.

Contains several organized structures called "organelles" (little organ). Each organelle takes part in a specific chemical activity.

Endoplasmic Reticulum
Definition
The endoplasmic reticulum is a system of double membranes that lie parallel to one another, fastened to both the cell membrane and the nuclear membrane.

Function
Acts as part of the cytoplasmic vacuolar system (along with other organelles) to allow materials to pass through cytoplasm.

Figure 3.

Applications for Teachers

Until the Macintosh is more widely used in schools, it may be necessary to view *ThinkTank 512* as primarily a utility for teachers. The program would be extremely helpful as a tool for creating lecture or discussion outlines. A feature allowing users to determine how many levels of the outline are printed can help teachers to produce both a speaking outline (just main topics and subtopics) and

© Copyright 1985 ICCE

a more detailed outline (including all headings and text) for prior review or quick reference.

ThinkTank could also facilitate the process of planning and organizing resources for a whole unit of study. Information from a variety of sources, notes as to which materials are needed for specific activities, even time estimates and field trip arrangements could be incorporated into an outline representing the grand plan. Sections could be printed as needed, e.g., daily lesson plans, questions for discussion, assignment guidelines for students, list of films to order, etc. After the day's instruction, ideas for modifying the unit (what worked, what needed more time, etc.) can be inserted where appropriate so that the unit is improved each time it's used.

For educational specialists, counselors, teachers and others who may need to keep anecdotal records on students, *ThinkTank* might find an additional use. Information on students can be arranged by topic, and updates easily inserted. Unlike most database files, there are no limits constraining the structure of information inserted and entries can be rearranged with a simple click of the mouse. Text windows help to keep confidential information out of sight and sections can be printed easily for conferences with district specialists or parents.

Problems and Limitations

ThinkTank 512 is an exceptionally well conceptualized program which runs smoothly and reliably. The only difficulty I had was in learning the meaning of various printing options. In an otherwise very specific and well-illustrated manual, the details concerning possible printing variations are not clearly presented. A bit of experimentation eliminated my confusion, but the publishers may wish to rework this section of the manual to maintain the standards of clarity exemplified throughout the rest of the product.

To increase the value of *ThinkTank 512* for educators, it would be extremely helpful to have a teacher's guide and possibly even a student manual. The former could contain suggestions for incorporating the program into classroom instruction and the latter might provide a tutorial and activity guide to get students creating their own outlines. Including sample outlines on the disk that demonstrate educational applications would also aid teachers in adopting this helpful tool. In addition, the publishers might consider adding another font size. The program currently offers two options: 9-point and 12-point. The latter is great for lecture outlines, agendas, etc. but the former is really too small to be useful (or even legible)—something in between would be better.

And finally, my only real criticism of the product is that it doesn't carry out the writing process to its logical conclusion. For ultimate utility and flexibility, *ThinkTank* outlines should lead to final papers and reports in traditional, professionally accepted formats. This might be accomplished by allowing outlines to be translated to a word processing program for final editing and formatting, or by including more printing options within the existing program. I suspect that either of these may already be possible (in which case I'd like to have details on how to do it) or the developers have a rationale for why this is not the appropriate use of an "idea processor."

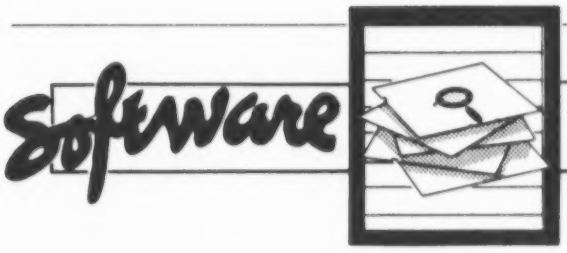
Recommendation

ThinkTank 512 is one of the best programs I've used in a very long time. It is easy to learn, operates reliably, and is accompanied by well-written documentation. It does what it sets out to do, and it does it smoothly. The program has the simplicity necessary to make it appealing to novice users and the flexibility to make it appropriate for a wide range of applications. There will probably be a time when similar outlining programs will be standard equipment on all word processors—until then, treat yourself to a *ThinkTank*.

Publisher's Response: The reviewer's request for transferring outlines to a word processor for final editing can be done by putting your *ThinkTank* outline on the Clipboard and pasting it into any popular Macintosh word processing program.

Living Videotext is now shipping *ThinkTank 512* version 1.1, which is compatible with Apple's *Switcher*, has new fonts and leader characters, new printing options, and adds outlining enhancements.

© Copyright 1985 ICCE



REVIEWS

Linda Rathje, Editor

TAKE 1

Baudville, 1001 Medical Park Drive S.E., Grand Rapids, MI 49506

Grade/Target Age Group: Four to adult

Subject: Animation, graphics, art

Cost: $59.95. Backup is available for $5, with proof of purchase.

Hardware Required: Apple IIc, IIe, II+ (64K minimum and Applesoft BASIC in ROM).

Recommended: Color monitor; second disk drive.

Reviewed by: Chris Clark, *Instructional Services Coordinator, Delaware-Chenango BOCES, Norwich, NY.*

Take 1 is a wonderfully designed animation program for both beginners and more advanced graphics fanatics. I recently taught an inservice course called "Visualization" in which teachers were given the opportunity to become familiar with several computer graphics production tools. *Take 1* emerged as a clear favorite of this group, which was a mixed bag in terms of subject areas and grade levels.

Take 1 can be used to produce and then view full-color computer movies. Easy-to-use, menu-driven modules are provided for drawing or modifying the animation shapes; creating scenes frame-by-frame including text, sound effects

87

© Copyright 1985 ICCE

and pauses; and combining the scenes to make a complete movie. All that is a bit complex, but only for the first few minutes of use.

Creating Actors and Actions

The most potentially time consuming aspect of the program is the creation of the "actors"—the individual pictures which, when viewed in sequence, will provide the effect of animation. But users are not required to start from scratch. The *Take 1* program disk supplies pictures and shapes which can be used to create actors, as well as a number of completed actors which can be duplicated or modified.

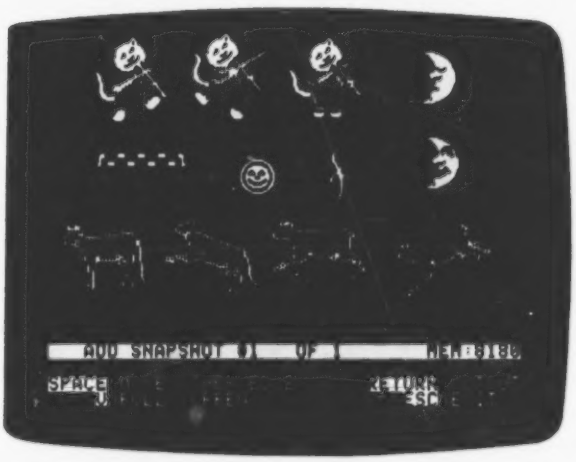


Figure 1.

In the Actors and Actions module an actor is made by selecting (clipping) individual poses one by one from a hi-res picture file (figure 1). Each pose is called a "snapshot," and snapshots may be copied and then changed to create new snapshots. A group of snapshots can be saved and used snapshot-by-snapshot, or several snapshots that together define movement (such as walking) can be given a name and saved as a pre-recorded "action." An actor called "cow," for instance, might have actions called "eat grass" and "jump over the moon." There is no shape size restriction as in some other programs; an actor can be as big as the screen.

For those who decide to make their own actors from scratch, a time investment of four to 10 hours can be expected, depending on how sophisticated the actor is to be and how much planning is done ahead of time. The first step is generally to use the Pictures & Background module to create an image containing different poses of the object to be animated. The Pictures & Background module is a sophisticated full-screen, hi-res picture editor which is almost worth the *Take 1* purchase price, even without the animation component. One may take parts of a picture and copy, cut or move them. It is possible to fill any enclosed areas of a picture with color, or edit the work dot by pixellated dot. And text can be added to the picture by using the hi-res text font available on the program disk.

Backgrounds are DOS pictures that are packed to take up less space on the disk than usual. Hi-res pictures and shape tables from other graphics programs can be used with *Take 1*.

© Copyright 1985 ICCE

Creating Scenes and Movies

Once actors and backgrounds are available, the Scene Editor comes into play. To create a new scene, an actor is loaded into memory along with a character set (if text is to be included in the scene) and a background. The user can then set a projection speed of one to 30 frames per second. Faster speeds provide smoother animation but slower ones make the same number of frames last longer. Individual frames of a scene containing text, actors, sound effects and pauses are combined, edited if necessary and saved as a completed scene. When there is more than one actor in a scene, the last actor cast will appear to be in front of the others. This overlay capability provides depth to the scene and allows for special effects.

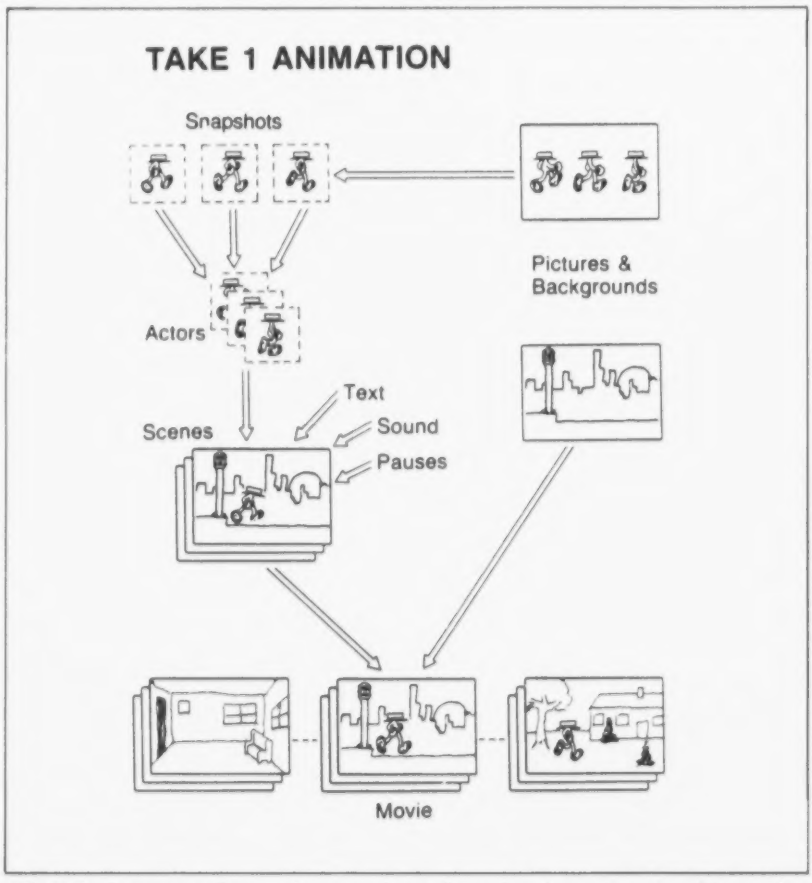
To make sure animation is synchronized, the actor from a previous frame can be superimposed on a new frame. When the scene is ready, a trial run can be made from the Scene Editor. New frames can be inserted at any point in the scene or old frames edited at a later date.

After scenes have been created a movie can be made. The Movie Editor lists which scenes and backgrounds will be in the movie and what order they will be played in. Each scene of a movie contains a background, a fade in and a fade out (17 fade options are available). Finally, the Movie Projector module allows viewing of the completed work as a single play or in continuous showings.

The *Take 1* Disk Utilities module is extremely helpful. It allows the user to format, catalog or back up a data disk; copy, delete or rename files; or convert pictures and shape tables made with other graphics programs. It will even check to make sure the actors and backgrounds a movie requires are on one disk. The disk files are neatly displayed by category, in alphabetical order.

Strengths

The program is designed for maximum ease of use. Help screens are available at a number of points. If an error is made while editing a picture, actor or frame, the last step can be undone by pressing the letter "U." Systematic double checks and warnings prevent hasty users from making any number of careless mistakes.



The Take 1 *sequence for creating an animated movie: a) create actors, b) combine them in frames, c) add text, sound, pauses to create scenes, d) combine scenes and background to make the movie.*

The 80-page manual is thorough and well laid out. It begins with an explanation of the different terms used (actor, scene, etc.), then holds the user's hand as s/he "makes a scene" using a prepared actor. After the basic operations are covered, the tutorial moves on through more tricky techniques. There are cute illustrations featuring "Li'l Doog'n" (one of the actors supplied on the program disk), useful screen photographs, and charts explaining each module. Tips and special techniques are suggested throughout the manual and a complete glossary is included at the end.

Actors and shapes are provided so that a minimum of time is required for a first effort at animation. Baudville also sells three supplementary animation library programs containing professionally designed animation art: "Actors and Actions" (in my opinion, the best), "Heroes and Villains" and "Business."

Disadvantages/Weaknesses

There are few weak points in the package. Movies can't run independently of the program disk. However, a *Programmer's Tool Kit* is available which allows users to create their own movie disk which will boot and run movies automatically without the *Take 1* program disk.

An annoying wait occurs between the program modules. Perhaps two or more could be merged, such as the Actors and Actions module with Pictures and Background, to reduce the delay and further ease program use.

Without taking away from the *Take 1* still picture editor, it's a shame there isn't a single disk (e.g., a "supergraphics" package) with an even more full-featured picture editor like *Blazing Paddles* or *MicroIllustrator, and* a complete animation utility like *Take 1*. However, publishers are having great success with

© Copyright 1985 ICCE

graphics programs and there may be a problem with disk space limitations.

Classroom Applications

The program has a wide variety of possibilities for the clever user. In creating illustrations for science classes, computer animation provides a break from blackboard and chalk (and computers don't generate dust). Text can be animated for use in the language arts or any other text application, and art and graphics classes can easily integrate *Take 1* into their curricula. Special projects or presentations take on new dimensions with the addition of animated computer graphics.

And through the use of special interfaces, such as the Adwar Proc Mod board, computer graphics can now be videotaped on a home videocassette recorder. For students interested in media production, *Take 1* animation can be edited in with real-life action to add variety to videotape projects.

Recommendation

Take 1 is a flexible graphics utility program which affords sophisticated users and novices alike the opportunity to create amazingly smooth animation. No programming skills are required, it is reasonably priced and the documentation is very good. I highly recommend *Take 1* alone—and with one of the optional animation library disks it is even better.

Publisher's Reply

Regarding the "wait" between program modules:

A four- to five-second pause is very small when you consider that each program module is 20K to 30K in size. *Take 1* is loading these modules from the disk as fast as is physically possible. A great deal of effort went into creating a disk operating system that would minimize disk access time. If the system were running under Apple DOS, the disk access would be up to 10 times longer!

The physical size of the program modules also prevents combining them. Remember that *Take 1* is a very sophisticated system running in a 64K environment. It would require a computer of at least 256K to have all the program modules in memory at once.

© Copyright 1985 ICCE

SUPERKEY

Bytes of Learning, Inc., 150 Consumers Road, Suite 202, Toronto, Ontario, Canada M2J 1P9

U.S. Distributor: Scholastic, Inc., 730 Broadway, New York, NY 10003.

Grade/Target Age Group: Ages 8 to adult

Subject: Keyboarding

Hardware Required: Apple IIc, IIe, II+ (64K minimum); 1 disk drive; monochrome monitor

Cost: $69.95 incl. 1 backup disk; $295 for Classpack with 10 disks, 1 manual, 10 student booklets, plus teaching aids.

Reviewed by: Laura BeMent, Business Education Department Chairperson, Anoka Senior High School; Jim Hirsch, Mathematics Department Chairperson, Coon Rapids Junior High School; Randall E. Johnson, Ph.D., Instructional Technology Consultant, Anoka-Hennepin Schools, District #11, Coon Rapids, MN 55433

Description

Superkey is a personal keyboarding tutor suitable for age eight (third grade) and above. The package contains two disks, a 39-page spiral-bound manual and five reproducible key charts and lesson records.

The program has four main parts:

- Posture introduces correct body and hand positions using simple but realistic graphics;
- Finger Names introduces the home-row finger position on the Qwerty keyboard using animated graphics;
- Lesson contains 10 units of instruction to develop keyboarding techniques, illustrating every finger action; and
- Skill Check provides practice, assesses progress and suggests ways to improve skill level.

© Copyright 1985 ICCE

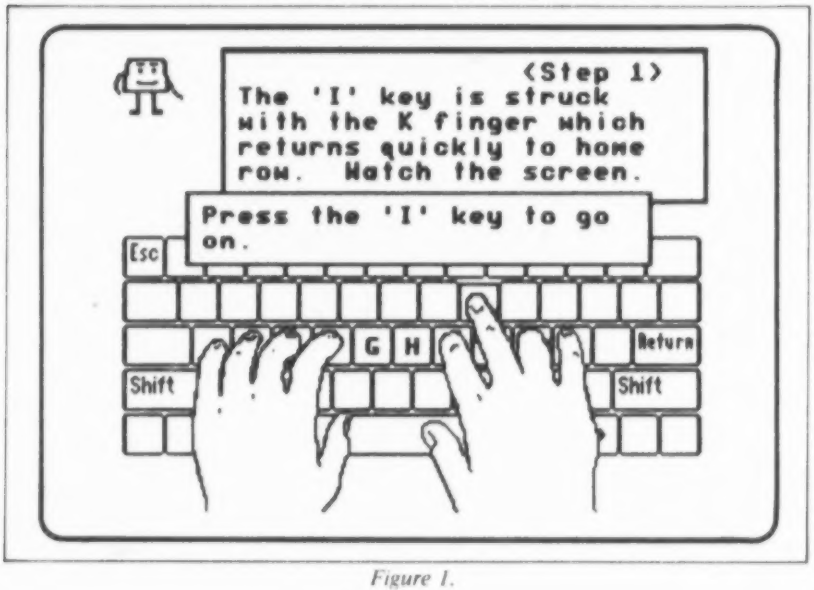


Figure 1.

Applications

Superkey has the potential for many individual, classroom or small-group applications. As an introduction to the keyboard for younger learners and first-time keyboarders, it would be most effectively used with instructions and demonstrations from a teacher. At the elementary level, beginning at third or fourth grade, we encourage a strong teacher role while acquainting students with this package and in monitoring the students' keyboarding techniques. Bad habits should be corrected as soon as possible so they are not practiced. For persons who have had some experience at the keyboard, including those who have acquired some unique and unorthodox approaches, the program offers a chance to "do it right" this time. A junior high student commented, "I found it helped me learn more with the right fingers."

At the secondary school level *Superkey* could be used in existing typing courses, especially when the keyboard is being introduced or for students who have been absent.

Using the Program

Lesson selection is fast with 128K since the program will "total-load." If a machine with less than 128K is used, switching among the four program parts requires accessing the disk. It is very easy to move among the four parts of the program, even to repeat a small segment of a lesson.

The manual offers several good suggestions for using all or part of the program and how the lessons can supplement or be integrated with the teacher's own materials. While the manual would be used more as a teacher's guide for elementary applications, it is written clearly enough for a self-instructional mode for older students.

Strengths

We were all favorably impressed by the program. The graphics relating to posture and finger position are very realistic (figure 1). The keys are introduced in about the same order as they would be in a beginning typing course. The skill checks and practice exercises are very compatible with recognized touch-typing techniques. The focus of the Finger Names module is clearly on correct positions and is not confused with speed.

Very little reading is necessary to follow the program screens. The picture menu and the overall graphics are well done and add to the ease of use and keeping of the student's interest. The manual, too, is very easy to follow.

The Skill Check section presents paragraphs which measure accuracy and speed more realistically than single line entries. One student commented, "[It] contained sentences to type instead of different words bunched together. That made it more interesting than others I've worked on." Once students have completed all of the lessons in *Superkey*, they can keyboard their own writing or additional paragraphs provided in the manual.

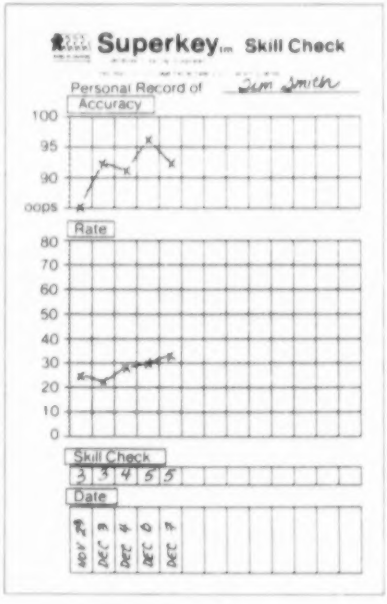


Figure 2.

Weaknesses

The only major weakness/omission we could find was the lack of a disk-based management system. While a chart is provided for keeping track of students' progress (see figure 2) as a substitute for such a system, monitoring the progress of many students requires each student to record his/her progress after each session, a practice which leaves much room for error.

Summary

We strongly recommend *Superkey* for a variety of instructional settings and would give it an "A" if it had a disk-based management system. As is, *Superkey* rates a "B+."

Publisher's Reply

It may surprise some readers to know that we believe the "people-based" management system featured in *Superkey* is the most educationally sound approach we could take. Rather than write

© Copyright 1986 ICCE

records to a diskette and subject students to arbitrary computer logic, we think it more appropriate and effective to leave the management of learning in the hands of the learner.

As a student operates the program, *Superkey* encourages the learner to make decisions about his or her next learning activity. Depending on the student's progress, *Superkey* suggests reviews of certain activities or invites the learner to advance to higher levels. At the end of each lesson and skill check, the student is prompted to record results on the *Superkey* record charts. Our field tests and a great deal of prior experience with educational computing confirm that students who are actively involved in managing their own skill development —as in the case with Superkey—have much more positive attitudes and greater interest in the learning at hand.

The reviewers have apparently found *Superkey* to be the excellent product we intended it to be. A great deal of time, care and expertise was called upon to make sure that *Superkey* would provide the effective keyboarding instruction that so many people have been looking for. The four months of field testing which took place before *Superkey*'s release and the six months of excellent post-release feedback confirm that *Superkey* really is a winner!

SPELLING WORKSHEET GENERATOR

D.C. Heath and Company, 125 Spring St., Lexington, MA 02173

Subject: Teacher utility for spelling and vocabulary development

Cost: $54.95. Backup disk included.

Hardware Required: Apple II series (48K RAM minimum); 1 disk drive; monochrome monitor, printer

Reviewed by: Gail Sullivan, second grade teacher, Washington Elementary School, and Teacher on Special Assignment in Computers for Eugene (Oregon) 4J School District

Spelling Worksheet Generator produces three types of spelling worksheets from lists of words entered into the computer. After typing in a list of words, you may choose to have the letters scrambled, have a cryptogram printed or have blanks inserted into the words. With the options available to control the difficulty level of the cryptogram and the fill-in worksheet, a teacher can produce a wide range of classroom materials to reinforce and review spelling lessons. The variety of worksheets enables the teacher to provide for all levels of skill in a convenient way.

A single keystroke after the title page brings up a straightforward list of printers (no parenthetic codes to confuse us) that work with the program. Those listed will print in double-width letters, resulting in worksheets which fill the page. Printers not listed will produce standard-width letters and fill only the left half of the page.

The main menu, the second screen, is explicit, offering five ways to proceed. Choose "INTRODUCTION" and you will see a concise program description, similar to the manual.

Choice 2 is "TYPE A LIST OF WORDS." Be ready with your word list —you will be entering words on the third screen! Few programs act with such dispatch. If they did, we would be meeting more of our colleagues at the computer.

Lists can include up to 40 words, and the disk will accommodate 120 lists. After entering a list of words, you may change the words or the order of the list.

After entering one list, which is automatically saved, you may enter more. This is an attractive feature: You may enter ten weeks' lessons and have your spelling assignments for a trimester transferred into the Electronic Age in short order.

The third option on the main menu, "EDIT LIST(S) OF WORDS," brings the names of the word lists stored on the disk to the screen. The first-time user

© Copyright 1986 ICCE

will find that some useful practice lists have already been entered. The options for editing are correcting, adding or deleting word(s) or list(s), retrieving additional lists after working on one, changing the order of the words, and finally saving the edited list with the same name or a new name. This option provides full control over the list(s), from minor to major changes.

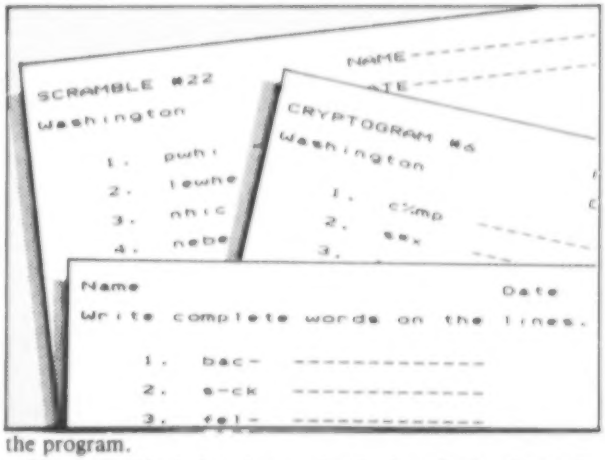
The fourth option on the main menu, "PRINT A WORKSHEET," presents a screen with these options: PRINT A SCRAMBLE SHEET, PRINT A CRYPTOGRAM, PRINT A FILL-IN. After selecting the type of worksheet, options for its heading provide an opportunity to identify the worksheet as well as to print directions.

In a cryptogram, a maximum of 15 letters in the puzzle can be replaced with symbols. The words may be printed at the bottom of the sheet if you choose. In a fill-in worksheet, blanks may comprise 20 to 80 percent of the words. Answer keys are available for printing in all three forms of worksheets.

The final choice on the main menu is "DELETE A LIST FROM THE DISK."

Strengths

A number of details lend strength to the program. The double-width printing helps create an attractive, balanced format for the worksheets. Upper and lower case letters are available. Upon entering an original list, saving the list is automatic. The computer reminds you with two beeps if you have edited a list in memory and not saved it on disk. Options for the headings and word banks on the worksheets, and the ability to adjust their difficulty level, increase the usefulness of



the program.

The manual includes a lesson plan for introducing the worksheets and nine printouts from the program to examine.

Weaknesses

The program contains all directions on screen. Although handy for the first-time user, the successive screens full of instructions are cumbersome for the regular user. The format of the choice screens should be streamlined from sentences to a list of items.

The keystrokes used to proceed through the program are confusing, since there is no consistency. For example, "Type 'N' . . ." allows you to continue one time, while on other screens "Type 'N' . . ." saves words, shows options or creates a new name. Because of this it is necessary to repeatedly read full screens.

These weaknesses in the program are not major roadblocks, but annoyances.

Summary

Spelling Worksheet Generator is a sweet little workhorse! The capacity of the program to store lists makes it possible to store over one year's spelling program on the disk. The procedure for entering lists makes it possible to transform your spelling program to electronic storage in the time it takes to key it, and it could easily be done at one sitting. The ease of producing worksheets provides the power to design materials to meet specific needs.

Spelling Worksheet Generator seems to almost invite users. Once grade-level word lists were entered, they could be accessed easily by all teachers at that grade level. Little time investment would be necessary to enable teachers to do this, and they could print their own selections.

Or should we keep the program disk in the Instructional Materials Center, and send a work order to the aide? "Spelling Worksheet Generator, Grade 2, Lesson 4, Cryptogram, eliminate all vowels, print key at bottom."

While white collar workers dream of the paperless office, I envision teaching without manila folders filled with dog-eared dittoes. *Spelling Worksheet Generator* is a step in the right direction!

[Spelling Worksheet Generator, written by Gregory N. Dubac, was originally published by Sterling Swift Publishing Company.]

Publisher's Reply: None.

© Copyright 1986 ICCE

Software REVIEWS

Linda Rathje, Editor

FrEdWRITER

SOFTSWAP, San Mateo County Office of Education, 333 Main Street, Redwood City, CA 94063

Authors: Al Rogers, *FrEdWriter* Paul Lutus, *Freewriter*

Subject: Word processor

Cost: $20, two disks, documentation and sample writing lessons included on disk.

Hardware Required: Apple IIe/IIc, modified II+; 1 disk drive; monochrome monitor; printer.

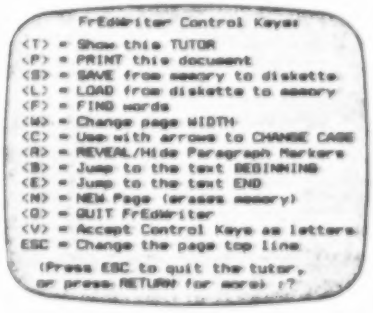
Reviewed by: Cathy Chmielowski Carney, Computer Lab Coordinator, Ryan Middle School, Fairbanks, Alaska

- FrEdWRITER
- EDUCALC
- CONCERTWARE+

FrEdWriter (Free Education Writer) is a public domain word processor available for a nominal fee from SOFTSWAP of California. Once you have the program you are free to copy and distribute it to anyone—but not to sell it. It is an enhancement of *Freewriter*, written by Paul Lutus of *Apple Writer* fame. Unlike much public domain software, its quality is as high as or higher than many commercial products.

The original *Freewriter* was a useful tool, but *FrEdWriter* incorporates significant improvements in its performance, documentation and educational usefulness.

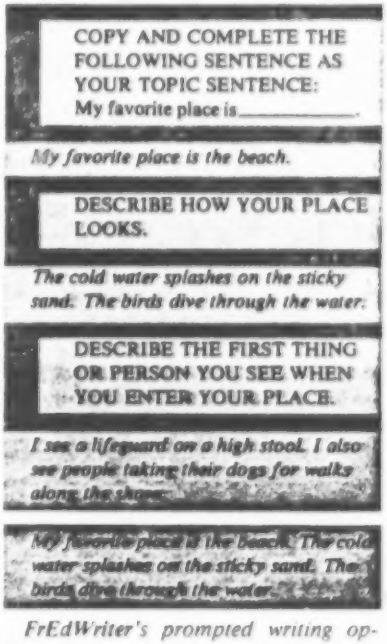
The technical features one would expect from a basic word processor are found here, accessed through control characters: control-S for save, control-P for print, etc. Critics of such systems may be mollified by the on-line help screen available to help remember these commands. Pressing control-T brings up a tutorial which explains briefly each control code, and which can be exited at any point. This, coupled with the mnemonic control characters, makes the system very palatable to users I know.


FrEdWriter Control Keys:
<T> = Show this TUTOR
<P> = PRINT this document
<S> = SAVE from memory to diskette
<L> = LOAD from diskette to memory
<F> = FIND words
<W> = Change page WIDTH
<C> = Use with arrows to CHANGE CASE
<A> = REVEAL/Hide Paragraph Markers
<B> = Jump to the text BEGINNING
<E> = Jump to the text END
<N> = NEW Page (erases memory)
<Q> = QUIT FrEdWriter
<V> = Accept Control Keys as letters
ESC = Change the page top line
(Press ESC to quit the tutor, or press RETURN for more) :?

Users of *Freewriter* will be delighted to learn that its major flaw has been corrected here; *FrEdWriter* will print from within the program itself. Furthermore, the writer can now choose to have the document printed to the screen first, to check for potential formatting errors. Those who have used a word processor with this capability can appreciate how much time and paper it saves. In addition to printing to screen, the printing options include such standard features as choice of line spacing; top, bottom and left margins; page numbering and heading; and choice of page from which to begin printing.

The program recognizes whether it is in an Apple II or IIe/c environment, and loads the appropriate version. Apple II users need 64K of memory (a language card) in order to use the program. It is also highly desirable to have a lower-case display adapter, but I have known people to use the program on a II+ without this adapter. They simply press the shift-lock (esc-shift) before writing, and accept the printout in all upper-case letters. On an Apple IIc, or IIe with 80-column card, text can be displayed in either 40- or 80-column format.

More sophisticated users who enjoy enhancing their writing through the various capabilities available on their printers will be delighted to know that *FrEdWriter* permits sending, at any point in the text, any number of control codes to the printer: foreign language sets, italics, boldface, or any other modification supported by their printer.


COPY AND COMPLETE THE FOLLOWING SENTENCE AS YOUR TOPIC SENTENCE:
My favorite place is _____.
My favorite place is the beach.
DESCRIBE HOW YOUR PLACE LOOKS.
The cold water splashes on the sticky sand. The birds dive through the water.
DESCRIBE THE FIRST THING OR PERSON YOU SEE WHEN YOU ENTER YOUR PLACE.
I see a lifeguard on a high stool. I also see people taking their dogs for walks along the shore.
My favorite place is the beach. The cold water splashes on the sticky sand. The birds dive through the water.

FrEdWriter's prompted writing option lets teachers create on-screen prompts or instructions for guided writing activities.

In addition to improved technical performance, *FrEdWriter* includes much more extensive documentation than *Freewriter*. There are four separate document files on disk, written by June Wedesweiler Dodge, which total 45

Copyright 1986 ICCE

printed pages. They explain and give examples of the program features and list step-by-step instructions for the novice user. The user may either print these on paper to have as a ready reference or examine the files on screen when a question arises.

Because the people modifying *FrEdWriter* were educators, they did not stop with simply improving performance and documentation. An additional enhancement—prompted writing—is specifically designed for educational use.

In prompted writing, the instructor can create boxes of prompts which cannot be changed by the student. For example, a teacher may wish to encourage writing by storing the questions: "Who is your favorite hero?" "What does this hero do that makes you admire him/her?" "In five years, how will you be like that hero?" After each question, the student is to respond in complete sentences. The space between the questions expands to accommodate the answers.

The final product may be printed with or without prompts. If printed without prompts, and if the student has answered the questions in complete sentences, the resulting printout will be a complete story or coherent essay. Once students are accustomed to using prompted writing, they can further extend their writing/thinking abilities by creating their own prompts to help other writers.

The program may have weaknesses, depending upon what a particular individual wants. There are a few menus and credits that appear each time the disk is loaded, and the impatient user may find this annoying. Experienced computer users will doubtless find some features omitted which are present in other word processors. The graphics and calculator programs that were part of the original *Freewriter* disk are gone, so if you have a copy of *Freewriter*, you may wish to keep it.

Recommendations

Anyone with an Apple computer and without a word processor should obtain this disk immediately. Even if you will not use the program yourself, it is wonderful to be able to give free copies to students or to colleagues who have not yet discovered the virtues of word processing. Those who already have a first-class word processor may not see the need for it, if their uses are simply correspondence and writing papers. But if they teach and their students have access to computers, they will find the prompted writing features very useful. Furthermore, being able to print to the screen and to access the printer through control commands are quite useful features not found in every word processor. For the price, it can't be beat.

Publisher's Reply

FrEdWriter's ProDOS user interface is an improvement over the original *Freewriter* in that it shields users from obscure and confusing demands to enter "prefixes" and "pathnames," making normal disk management easy for all users.

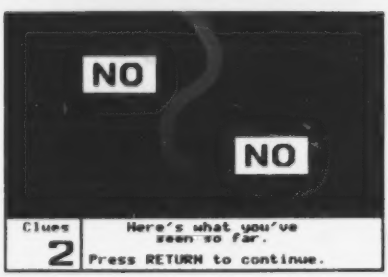
The final version of *FrEdWriter* (Version 4) has the following additional enhancements: automatic line centering, improved printing features, all new documentation and improved error trapping.

A Spanish *FrEdWriter* for the Apple IIe is also available. *Spanish FrEd* requires a character chip to display foreign characters on screen and comes with the purchase of a Spanish Wizchip from Hands-On Training Company, 4021 Allen School Road, Bonita, CA 92002 for $40 (quantity discounts available).

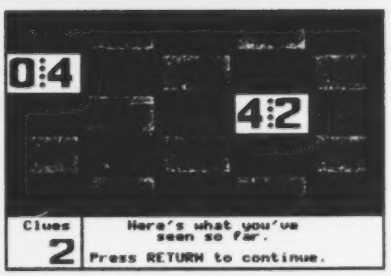
© Copyright 1986 ICCE

Software Reviews

Linda Rathje, Editor



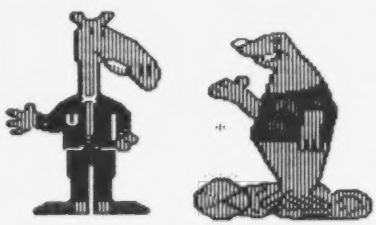
8:00 p.m.: A loon hides behind one of the 25 doors on the screen, but the one we select creaks open to reveal an emphatic, "NO, I don't see the loon!" Aha, so the loon does not hide in the box we chose or any of the boxes it touches. Hmmm, what box should we pick next to give us the best clue to where the loon is?



11:30 p.m.: Our door reveals two numbers, "4:2." On paper we sketch a five-by-five grid and mark our position. Counting horizontally and vertically we check each box that is four squares away then two squares further in any direction. A pair of rhinoceros hide in two of the checked boxes. A little more information and we can make our guess.

2:10 a.m.: Can it be that late? I've (not "we" anymore) been positive I had located the two llamas twice. Both guesses were wrong. I select "SAME PUZZLE" so the llamas will hide in the same boxes again. The clue in the box I choose is a "5," representing the sum of the distance to the closest llama plus its distance to the other. After seven clues I press "R" to review what I've got. Suddenly the pattern makes sense. Remembering how sure I was twice before, I check all the possibilities I can think of before I make my official guess. I'm semi-confident as I press the last "RETURN." Viola! Two llamas tip their ears in salute to my problem solving skills or my perseverence or both. Now, at last, I can go to bed!

3:50 a.m.: Now that I can find the llamas in nine or fewer guesses every time, I wonder if it's possible to get it down to eight?



Safari Search consists of 12 terrific puzzles, each of which can be done over and over again. The first six puzzles are "One Animal Safaris" (one animal to find in a five-by-five matrix), which differ by the types of clues given. In the first puzzle, "Intuit the Iguana," the clues are a simple "NOT HERE" for each incorrect guess. The iguana is usually hard to find and the strategy is unsophisticated (really just random guessing). The second game, "Find the Flamingo," features HOT, WARM and COLD clues, and if the clues are properly interpreted, the flamingo is easier to find than the iguana was. In the remaining four One Animal Safaris the clues become progressively more thought provoking and interesting.

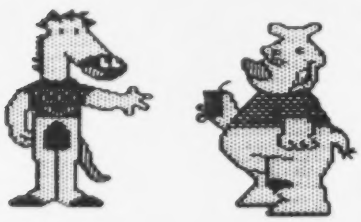
The next six puzzles are more difficult "Two Animal Safaris" (two animals in the matrix). The clues in these are variations of the clues given in the one animal safaris, but the extra animal makes their interpretation much harder.

© Copyright 1986 ICCE

Strengths

Safari Search is an excellent program. It is unique, well crafted, glitchless and an ideal program to use to help children develop skills in solving problems that require elimination of possibilities and interpretation of clues. Some of the reasons it excels:

- It is self-motivating, containing just the right elements to get kids excited and the variety to keep them that way. The option to change games keeps interest level high even for those with short attention spans.
- It encourages positive interaction among students as they struggle to solve the puzzles. There is no competition, no real score (though the number of clues is tallied), and wrong guesses aren't catastrophic because the same puzzle can be done again and again.
- Tactics learned in one game can be used to help master the next.
- There is no one "right" approach to the puzzles. In fact, discovery of alternative strategies is one of the program's most worthwhile challenges.



- *Safari Search* is a good program for those who usually have difficulty with puzzles. The first few games are short (three to 10 minutes) and relatively easy, allowing students an opportunity to develop confidence in their ability to solve these problems.
- Instead of visually overloading us with a screenful of clues, only the last clue is displayed. But when a review of all the evidence is desired, a simple "R" will put all the clues given so far on the screen.
- Screen prompts guide users effortlessly through the choices; the manual is not required.
- Control-E allows restarting the same puzzle or selecting a new one anytime.
- The program does not provide the answer after a wrong final guess; the student is given a chance to repeat the puzzle again (with the animals hidden in the same places).
- The program "reads" whether an Apple II + or newer model is being used and adjusts the screen prompts and keyboard functions for cursor movement accordingly.

Weaknesses

Though they should not keep the teachers from using the program, there are a couple of problems with *Safari Search*. It is not possible to view the completed puzzle after the final guess. If students could discuss what happened during their last game while viewing the final screen, they could examine their mistakes and correct guesses.

Instructions cannot be reviewed after the puzzle has begun. Students opting not to read instructions in the beginning (90 percent of my students) must exit to the main menu and start over to get help.

Using the Program in the Classroom or Lab

I found demonstrating the first puzzle, "Intuit the Iguana," a good way to introduce *Safari Search*. As students made guesses, they began to hanker for clues (all they got was a "NOT HERE" for each incorrect guess). In my classes of first through fifth graders, I had students brainstorm the types of clues they would find helpful or fun. They named most of the ones in *Safari Search* and were very anxious to try those puzzles with clues they had suggested.

Minimal instruction is required. I demonstrated the mechanics of *Safari Search*, gave some tips such as the usefulness of "R" to review clues, the value of drawing a grid on scratch paper and a system for taking turns, and then got out of the way.

As I circulated among groups of two or three students, there was excitement and almost universal involvement. The program's ease of use kept problems and mechanical frustration to a minimum, and careful rereading of instructions solved the few problems I saw. Incorrect answers did not seem to quell the enthusiasm; one student's musings went like this: "It must be that one! Argh!

© Copyright 1986 ICCE

Wrong! Oh, I get it! It could be this one too . . . or this one. Let's see what the clue is here. . . . Ah, so it must be there! . . . All right! This is great!"

I found post-game debriefing effective. In my classes, discussion centered on strategy: "Is there a best first guess for this puzzle?" etc. This seemed to help random guessers and puzzle-phobes develop puzzle-solving systems. After several sessions, analogies to searches for lost mittens and ways to improve an unsatisfactory recipe helped focus problem solving skills on real life problems that can be solved by eliminating possibilities.

Conclusion

Safari Search helps students experience inference, improve intuition, practice deductive reasoning and determine how to eliminate possibilities, thus developing important problem solving skills. In addition, the program is fun, fosters cooperation rather than competition, and requires little teacher attention after initial instruction. It is well crafted, comes from a reputable company and is reasonably priced. When I cry out for better software, this is what I have in mind. I highly recommend it.

SUPERPLOT

EduSoft, P.O. Box 2560, Berkeley, CA

Grade/Target Age Group: Grade 8 through adult

Subject: Math

Cost: $49.95; $35.00 site license (authorizing multiple copies or networking at a designated school site; five additional copies of documentation)

Hardware Required: Apple II family (64K RAM minimum); 1 drive.

Reviewed by: Dr. Ronald Steffani, Chairman, Mathematics Department, Southern Oregon State College; Judi Mathis, Oregon Math Education Council, University of Oregon

As a math teacher, you may have overheard some of these comments made by students whenever simultaneous equations are covered:

"It's okay if the answer lands on good numbers like (5,10)."

"Yeah, I have trouble with the fractions."

"I hate it when they intersect off the page like (150, −600), like who cares with an answer so far away."

"I just get tired doing so many—I still don't know what is going on."

These students do not need more homework or a computer program that is cutesey and fun but may cloud more than clarify. These students need a tool that helps them graph to such a degree that they begin to understand the process and the usefulness of solving simultaneous equations.

SuperPlot is a function-graphing program for these and other students designed to graph any polynomial, trigonometric, logarithmic or exponential function. It quickly plots up to five graphs, which can be displayed in different colors, clarifying the presentation. Graphed functions can be zoomed, scrolled and expanded for investigating intersections, asymptotes or other key areas of interest. While the users guide is clear, the program is so easy to use that it need be read only once.

Program Description

The program begins with Enter a Function. With a simple return it is ready to be a work horse. "Y =" appears, ready to accept an equation, which takes a minimum of keystrokes. Functions are entered using standard algebraic notation, but without entering multiplication signs, exponent carets or closing parentheses. One or two letters are used for entering conventional functions (e.g., S for sine, N for ln, and P for pi).

Once a function has been entered, the See Graph option displays a graphic page on which a pair of axes is drawn and functions are quickly sketched as a series of dots. Once all the functions have been sketched, axes labeled (with default limits of x = −10 to 10, y = −7 to 7), and tic marks drawn, the functions are drawn in detail. The color option and a color monitor provide discrimination between curves. With the sound option on, short bleeps change pitch as each function is graphed, up and down with the plotting, reinforcing the image of a cyclic sine curve or an exponential runaway. Sound and color can be toggled through the Changing Graphing Mode option.

Next the interesting features can be explored. The graph can be manipulated by scrolling right, left, up or down to view different portions of the functions. Zooming in to view a region of intersection or expanding to find a lost function can be done on either axis independently or both simultaneously. Each zoom leaves the origin fixed and halves the X- and Y-axis limits, thus showing a smaller portion of the graph, magnified by a factor of four. Expanding is the inverse of zooming. All options can be repeated and intermixed in any order to focus on the problem in question. An available help screen details necessary keystrokes.

Even without zooming or expanding the limits are adjustable. If the default values for the X- and Y-axis are not suitable for the intended function, they can be changed through the Set New Limits option. A menu offers a choice of default values, e.g., Trigonometry limits $(0,2\pi)$ and $(-2,2)$, or user-defined limits. New default limits can be saved for domain and range.

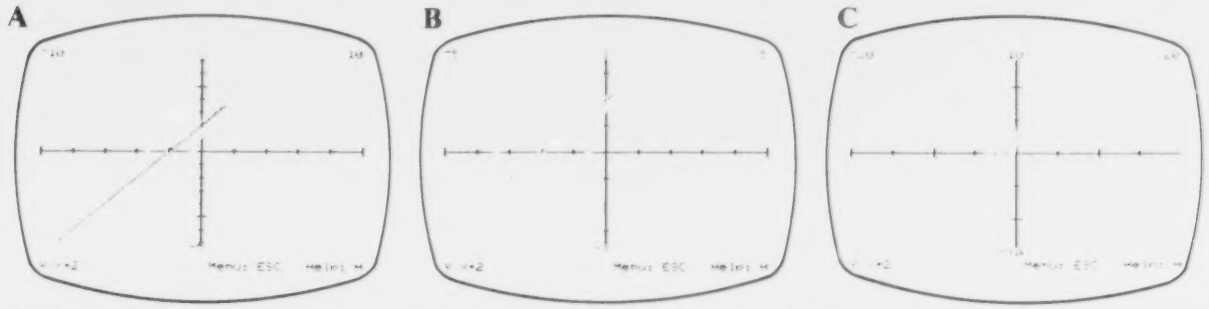
Ready to begin a new topic or different graphs? Erase and Erase All options allow one or all of the graphed functions to be removed from the graphics screen. Demonstrate one set of perpendicular lines, have students discuss any similarity in the equations of the lines, then add another set of perpendicular lines in just a few seconds. Have the students design the next pair of lines that might be perpendicular or a second perpendicular line to one of the existing lines. Presenting any family of functions all on the same graph enhances the concept attainment for that family.

When the dot mode is on, functions will be graphed as a sequence of dots. If there is a vertical separation between dots, no line is drawn connecting them. With the dot mode off, lines will be drawn connecting each dot. In this mode the sine function will appear: The greatest integer function will look like a stair step.

Strengths

SuperPlot fills the need for a teacher's demonstration tool in graphing. Teachers can face students and facilitate the discussion while the graphs are

© Copyright 1986 ICCE



The graphs of the functions $y = x^2 + 4x + 2$ and $y = x + 2$ can be viewed using the default limits for the axes (A), zoomed (B) or expanded (C).

being displayed, not have their backs to them, plotting points and talking to chalk.

There are many graphing programs, so why use *SuperPlot?* Because the program is clean. There are neither bouncing animals nor too many grid lines on the screen to complicate viewing. It even avoids the time-consuming start-up logo too often suffered through when loading a program. The minimization of keystrokes reduces opportunities for typing errors, eases demonstrations and expands the number of keyboard users.

Graphing is done quickly and accurately and is visible from a distance. The zoom, expand and scroll options are easy to use and allow teachers to design good student-centered discovery lessons for examining families of related functions.

Weaknesses

Two weaknesses exist, each of which limits the use by the classroom teacher. Once a set of functions has been graphed, manipulated and explored, "snapshots" of the work are desirable, but there is no printing capability built into the menu. Second, this graphing program, like others, fails to address the ever-elusive inequality. Life so rarely functions as points on a curve, distinct and locatable. It is more often like the bank book that is balanced above some minimal limit.

A minor difficulty can occur through the one- or two-keystroke entry of some functions. However, after getting "sin(int(ln(" when attempting to enter the word SIN, one learns fast. The graphs could be drawn with clearer resolution if more points were plotted, but this would slow down the plotting process. It would also be helpful if horizontal and vertical asymptotes could be drawn when needed. An "out of domain" message would be appropriate when attempting. In(0) or any other values for which a function is undefined.

Applications

SuperPlot is designed for use at many grade levels, even grade school, if students understand the notation and concepts of graphing. This electronic grid is a teacher's demonstration tool for introducing, clarifying and reinforcing the effects of changing function parameters, roots of an equation, points of intersection, critical points and intersection regions.

An even more exciting use of the package would be teacher-designed exercises where students, working in small groups, could discover concepts and generalizations about functions and their graphs. Estimation skills are emphasized with this visual medium even before students have the symbolic manipulative skills to solve for the roots.

Recommendation

We recommend that teachers from middle school through college make use of this easy-to-use graphing program as a valuable demonstration tool in their teaching and a valuable discovery tool for their students. The use of this software will take the drudgery out of graphing complex and unusual functions and allow students to explore topics formerly thought too advanced for them.

Maybe next year students' comments will sound more like:

"Oh, yeah, it's easy to find the intersections."

"My guesses are always a little short, but I'm getting better."

"It's like detective work to find the intersections when they are off the screen."

"I liked doing that simultaneous equations stuff."

Publisher's Reply

Thank you for the excellent and thoughtful review. We are very proud of *SuperPlot*. We plan to provide the printing capability suggested by the reviewer, and will offer the upgraded version to current users for a nominal fee.

Copyright 1986 ICCE

MASTERTYPE'S WRITER

Scarborough Systems, Inc., 55 South Broadway, Tarrytown, NY 10591

Grade/Target Age Group: Grade four and above

Subject: Writing

Cost: $69.95 Apple version; $44.95 Commodore
Publishers will mail at no charge a back-up disk to purchasers returning the warranty card. Damaged disks will be replaced after the 30-day warranty for a $5 fee. The program Tutorial may be freely copied.

Hardware Required: Apple IIe (128K) or IIc, Commodore 64 or 128

Reviewed by: Cathy Chmielowski Carney, Computer Lab Coordinator, Ryan Middle School, Fairbanks, Alaska

Word processors abound, so what makes *MasterType's Writer* different? I found four intriguing features:

- It uses color to "filter" (hide) text;
- It has two adjustable text windows;
- It permits font-editing; and
- It allows sorting of numeric and alphabetic lists.

Working with Color

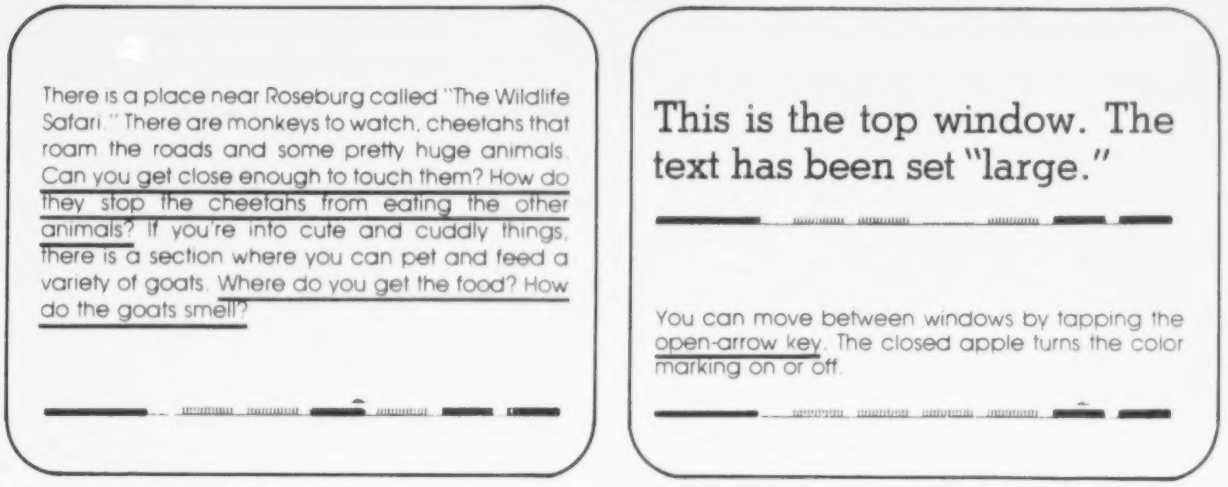
MasterType's Writer provides a color marking feature for identifying text to be displayed or hidden from view. The program's text filtering capacity works best with a color monitor, but it also works adequately without color, using variations in screen patterns to signal different colors. Selecting from the five color bars at the bottom of the screen, students can underline text as they type or underline text already on the screen. Once marked, selected text may be hidden or shown on the screen or printer at will.

This suggests a number of possibilities. A teacher could create writing activities with on-screen prompts which are hidden when the student prints the composition. When reading one's own or another's work, comments could be added using a different color, then either printed or not. An outline in one color could be printed alone or along with its text. Students working in pairs could each answer the same questions without seeing each other's work on the screen, then reveal both answers together for comparison and possible combination.

Using Text Windows

The program also has the ability to partition the screen into two windows, each of which can be edited, formatted, saved and printed separately. I found this feature handy for viewing my outline in one window and writing in the other. Whenever I wanted to see a larger section of my outline or to view my writing on the entire screen, I adjusted the boundary between the two windows. It is possible to change, at any

© Copyright 1986 ICCE


There is a place near Roseburg called "The Wildlife Safari." There are monkeys to watch, cheetahs that roam the roads and some pretty huge animals. Can you get close enough to touch them? How do they stop the cheetahs from eating the other animals? If you're into cute and cuddly things, there is a section where you can pet and feed a variety of goats. Where do you get the food? How do the goats smell?
This is the top window. The text has been set "large."
You can move between windows by tapping the open-arrow key. The closed apple turns the color marking on or off.

MasterType's Writer includes color marking for identify text to be displayed or hidden (A), and adjustable windows (B).

time, the size of the windows in relation to each other, or to move between windows with a single closed-apple keystroke.

There are other ways to use these windows, too. New ideas, tangential to the main topic, often come while one is composing or revising a piece of writing. Instead of losing these ideas, a user can regard window two as a notepad, jotting down ideas to be saved in a separate file and elaborated on later.

On-screen Typefaces

Another notable feature of this word processor is its special fonts. Students can have their writing appear on screen in one of seven different font sizes, ranging from 70-column type to very large letters suitable for young children or visually handicapped, to script. In addition, new fonts may be created. Children who have enjoyed editing the characters in *Fontrix, Gertrude's Secrets* or *The Printshop Companion* will find the same satisfaction here. Some people who prefer a traditional writing tool over a typewriter or word processor because they like the freedom to create their own individual letters may find their styles supported here.

It is true that, no matter what the type style on the screen, standard characters will usually appear on the paper printout. But this program permits using the Imagewriter and a number of other printer and interface cards to print these fancy fonts in the form of a screen dump.

Sorting Text

One unusual feature in this price range is the inclusion of a sorting feature; paragraphs can be sorted numerically or alphabetically. I found this useful for quickly rearranging several paragraphs I had written. I simply numbered each, then selected Sort from the menu. In addition, material not in paragraph form (e.g., sections of an outline) can be identified as blocks of text by using special markers to designate the beginning and end. Treated as "records," they can also be sorted. Keeping an address file or list of names sorted this way would be simple.

Besides the above special features, the advertising touts this program as "easy to use," and it is. An inexperienced computer user can simply insert the disk, turn on the computer and write. The program recognizes and automatically formats blank disks used to save files. Since menus are accessible at the bottom of any screen by using the Escape key, there is no need to memorize control key sequences.

For the experienced user there are other options available. A number of menu choices can be accessed with control-key functions as well as menu selections, so frequent users need not always call up the menus. Files can be transferred to other word processors (e.g., *Bank Street Writer, AppleWorks*) by using a menu option to convert the *MasterType's Writer* file into ASCII form, and ASCII files from other programs can be transformed into *MasterType's Writer* files. A sophisticated user can define Macro keys that then become available each time *MasterType's Writer* is loaded.

Interactive Tutorial

For a beginning user, a most important feature will likely be the tutorial on the reverse side of the disk. It partitions the material into several easy-to-digest lessons, including how to delete text and switch between windows. At the beginning of each lesson, a few objectives are stated. After a short instruction, a review and quiz consolidate the learning. The sentence structure, vocabulary and large type permit even a young learner to follow the directions. Many of the lessons include a simulation of the word processor screen, so the user becomes familiar with it right away. The lesson which teaches the concept of word wrap —an idea newcomers to word processing often find initially confusing—is particularly good.

Both beginning and more experienced users should find the documentation helpful. The first few pages tell how to get started right away. The rest of the 100-page manual is arranged alphabetically, like a dictionary, with lots of cross references. The entry "Memory" has a paragraph of text, followed by the direction, "See also Erase, Files, Load, Resave a file, and Save." The manual ends with an index and an appendix describing the meanings of error messages.

© Copyright 1986 ICCE

Program Limitations

With good documentation, a friendly tutorial and a number of special features, is this the word processor for you? It depends.

In order to preserve ease of use, some flexibility is sacrificed. For example, there is little control of printer margins or line spacing. The program provides for only four global margin and line spacing parameters that can be saved as Setup specifications. Business teachers would need to carefully examine the limited range of formatting features available with *MasterType's Writer*. On the other hand, for most home and school use, this limited formatting flexibility may not be a serious handicap.

The text files are compressed and saved in a special format. In order to use these files with another word processor, such as *AppleWorks*, or to transmit files with a modem, they must first be converted to ASCII, using a menu option. If you will need ASCII files for either of these reasons, be sure you can make this conversion, and that you know how much time it will take, before buying this word processor.

The limitation I found significantly bothersome was not being able to move the cursor away from the bottom of the screen. In order to adequately revise a section, sometimes it helps to see what is below as well as what is above the cursor position. This is not possible, though, as the cursor always remains on the bottom line.

And the Apple version of this program only operates in a 128K IIe or IIc environment. This makes it useless to IIe users without the additional memory.

Recommendations

Elementary schools frequently have color monitors and may enjoy this program because it uses color to manipulate text. For use with young children, the enlarged type could be a deciding factor. If there is enough class time and computer access to allow children to experiment with creating and editing fonts, this feature is attractive. Some elementary teachers may use it because it can sort spelling words, student names or numbered test items.

With upper-grade students, using the second window for outlines or as a notepad for jotting random ideas while writing could be appealing. Using color for marking and filtering text permits students to comment on their own or each other's writing and could contribute to a rich process-writing environment. A print-to-screen (view) function—not available on all lower-priced word processors—saves on paper and time in formatting the final printout and could be particularly useful in schools, where there are often not enough printers for each computer and students need to be sure their writing is formatted properly before using printer time.

This program is also available for both the Commodore 64 and 128, though we did not review it. Owners of Commodores, who lack the array of choices available to Apple users, may be particularly glad to know about this inexpensive word processor.

There are many word processor programs available for basic writing and revising. For teachers looking for ways to further assist their students in improving their organizing, thinking and writing skills, I recommend examining the special features available in *MasterType's Writer*.

Publisher's Reply: None.

SCHOLASTIC LIFE SCIENCE DATA BASES FOR pfs: FILE

Scholastic Software and Targeted Learning, Inc.
Scholastic, Inc., 730 Broadway, New York, NY 10003

Grade/Target Age Group: Grade six through 12

Subject: Life Science, Data File Management

Cost: $99.95 to educators. Includes three identical sets of disks: Each set contains two disks, one containing two files, the other containing one file.

Hardware Requirements: Apple IIe or IIc (64K RAM minimum); disk drive; monitor; *pfs: File* disk. A *pfs: Report* disk, printer and second disk drive are needed for many of the activities.

Reviewed by: Dr. Patricia McClurg, University Lab School, University of Wyoming, Laramie, WY

Introduce your students to the *Life Science Data Bases for pfs: File* and you'll find classroom conversation elevated to differing predictions about the influence weight variations might have on an animal's body temperature and heart rate; or an argument over which class of animals would be most likely to have just a ganglion for a nervous system; or a consensus that cold-blooded animals wouldn't be found in the Arctic.

This package, designed to be used with *pfs: File* and *pfs: Report*, provides opportunities for students to manipulate data bases. My students used this tool to test hypotheses ranging from a predicted relationship between length of maternal care and life span, to the disappointing confirmation that man ranked number one as most common enemy among other animals.

Program Description

The *Life Science Data Bases* package is a series of five science activity units. Three units involve ready-made data base files:

- The "Animals" data file contains records for 132 animals, including information about weight, trophic level, heart rate, habitat, biome and reproduction.
- The "Biological Systems" file contains records for 92 animals, including descriptions for the skeletal, digestive, respiratory, excretory, nervous, reproductive, circulatory, muscular, sensory and endocrine systems as well as information about phylum, class and system organization level.

© Copyright 1986 ICCE

- "Bird Migration" contains records for 234 birds, including migratory status, arrival and departure dates, habitats and major food sources for birds that visit a specific area in Michigan.

The last two units in the manual detail directions for creating data bases on drugs and wild flowers along with suggested activities for searching, updating and analyzing information contained in these files. Support materials in the manual include a Quick Guide summarizing important information about using the file, a Mini-Reference detailing procedures for using the *pfs: File* and *pfs: Report* packages, a glossary, and appendices explaining possible error messages and other trouble spots.

I admit to being a bit overwhelmed by my first glance at the 285-page manual, but as I worked through the suggested activities, the manual's value became apparent. The five curriculum units are well-written tutorials providing carefully sequenced activities designed to teach students to use data files. Specific instructions in the early lessons help ensure success while later lessons encourage more challenging projects. Suggested questions for investigation and student worksheets for planning reports and printouts are provided.

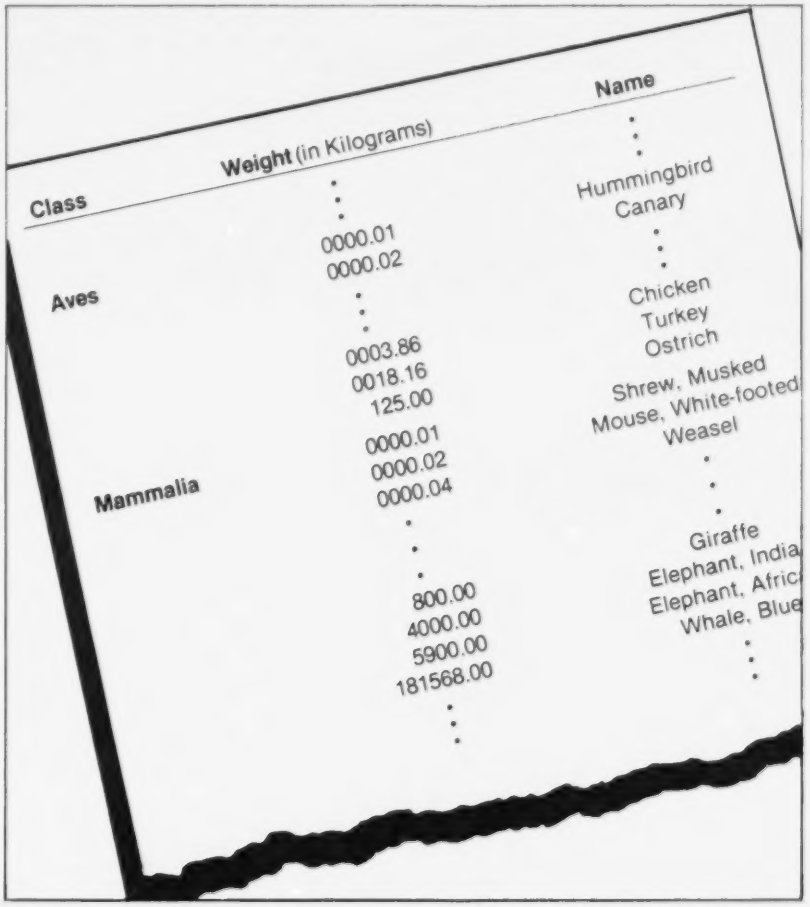
Classroom Observations

The "Animal" unit complemented our study of vertebrates, so we began with it. A few preliminary questions revealed that none of my students had searched data files, and that many were having trouble conceptualizing the idea of a computerized data base. I found that a few introductory activities (e.g., sorting and selecting baseball cards) provided concrete models which helped alleviate this problem.

I introduced the package using a classroom monitor for demonstration while students worked at computers in pairs. We spent the first week in the computer lab with students working through activities from the manual. Students learned how to manipulate the data files and search and sort the data to retrieve specific information. For example, they located animals that care for their young more than 216 months and then compared this group to animals that care for their young less than 12 months. In one activity students set up their own retrieve specifications. The combinations tried were varied, including searches for mammals with a weight greater than 10 kilograms and for animals who layed soft-shelled eggs but were active all year.

A subtle rise in the sophistication of investigations occurred when students began comparing data via *pfs: Report* printouts. The report program allows students to sort and print data in the first column by alphabetical or numerical order or by keyword. Data in column two can also be subsorted for repetitions of data found in column one. Additionally, calculations can be performed on data in the file and the result displayed in its own column. For example, the program can subtract data in column three from data in column two and print the result in column one.

As a group, we worked through the procedures necessary to complete several investigations suggested in the manual. For instance, we examined the question, "Which class of animals has the largest range of sizes?" by printing the class in column one, the weight in column two and the common name in column three. Examination of the sorted columns provided an obvious answer.



© Copyright 1986 ICCE

Next, I had student pairs investigate questions independently. The majority chose questions from suggestions found in the manual, but a few researched original questions. Before getting a turn at the classroom printer, students had to get approval of their projects by turning in a completed plan (worksheet provided in the manual). In two weeks' time all groups had completed their investigations and were prepared to share their results during a class "seminar." Sample questions researched included: Is there a relationship between an animal's litter size and its weight, life span and biome? Do animals whose young develop internally care for their young longer than animals whose young develop externally? Do poikilotherms have different enemies than homeotherms?

Considerable excitement was generated when students discovered patterns. This excitement grew when we began searching the "Biological Systems" data base for possible similarities and differences to the human systems we were studying in detail.

One search in particular created a stir. The question, "Do all animals with a kidney also have a bladder?" resulted in various predictions. When the printout started it became apparent that fish had a kidney but not a bladder. This made sense and my students were confident that they had determined the reason. Fish live in water—they don't need a storage chamber for liquid waste. Confidence changed to uncertainty when the birds began to appear. Another search was on, this time including reference books as well as the "Biological Systems" data file. The students were determined to reconcile this "discrepancy."

Currently, we are completing the activities suggested in the drugs unit and are learning to build data files. Each student is responsible for collecting data about four drugs. In addition to the resources suggested in the manual, we have tapped the expertise of local doctors and the school nurse.

After several heated discussions, students are learning the limitations imposed upon data bases by the selection of keywords. Also, they are learning—the hard way—the importance of validating data entries. One student, positive that the computer had lost his entry, searched through all the entries and discovered a misspelled word in his.

Problems Encountered

Although manipulating any data base requires learning specific commands, one *pfs* command caused confusion: Control-C is required to move from screen to screen. My students, used to pressing either the return key or the space bar to continue, had trouble remembering the Control-C command. A more serious problem appeared when students pressed shift-C or just C and inadvertently changed entries in the data files. This made later searches on these entries impossible until the data file was recopied from the backup file, or the change was noticed and reverted.

Another difficulty arose when we wanted more than one condition per keyword. For example, when trying to locate animals that had kidneys but no bladders we had to try numerous techniques to obtain such a sort. We finally settled for "not . . . kidney bladder" which selected extraneous files as well. A simple "'kidneys' *and* 'not bladder'" condition would have been useful. And as with any data base, the usefulness of the data and the types of questions that can be researched is limited by the information in each file.

The five units chosen correlate with topics covered in most introductory life science texts; my middle school science class made extensive use of the "Animals" and "Biological Systems" files. But while the animals chosen were fairly representative, I would have liked a more extensive file. I was also disappointed in the limited geographical region reported in the "Bird Migration" file.

I found that realistic management of this program required extensive hardware and software resources, a factor for real consideration. Time allotments of 35 to 50 minutes on the computer were necessary for most activities. Access to a computer lab in the beginning stages was very helpful, but it necessitated multiple copies of all programs.

Using this package requires *pfs: Report* and *pfs: File*, which are not included. While instructions for making backup copies of the data files are given in the manual, *pfs: Report* and *File* are copy protected.

I found printouts essential to the ef-

fective use of the package in a classroom situation, so ready access to a printer is necessary. Since we only had one printer available, *pfs*'s slow sorting time increased the wait between users.

Recommendations

The *Life Science Data Bases for pfs: File* package provides a model for extending scientific inquiry into data bases and reaping results not easily realized with other techniques. My students discovered interesting relationships among the characteristics and environments of living organisms and investigated questions more global in nature than those researched through more traditional activities.

Given the expense involved in securing data base management programs, I would like to see Scholastic publish versions compatible with other widely used systems. I believe this package provides quality examples for computer application courses and is valuable as a tutorial for teachers who wish to integrate data bases into their teaching. Life science teachers interested in using these materials should preview the data files for content compatible with their course objectives. Be aware that an investment of teacher and student time will be necessary to learn how to use the materials, but the return on this investment will be a useful classroom tool.

Publisher's Reply

None.

MECC INFORMATION MANAGER

Minnesota Educational Computing Corporation, 3490 Lexington Avenue North, St. Paul, MN

Grade/Target Age Group: Grades seven through 12

Subject: Problem solving simulation using productivity tools

Cost: $49.00

Hardware Required: Apple IIe or IIc (64K RAM minimum); single disk drive; two disk drives and printer recommended.

Reviewed by: Nancy Parks Sopp, Computer Lab Teacher, Fairbanks North Star Borough School District, Fairbanks, Alaska

Information Manager simulates an environment in which students use computer-based tools to perform some of the tasks involved in operating a rock-and-roll radio station, MECC-FM. The tools include an electronic mail system, a word processor, a data base and a calculator. Students should already be familiar with one or more computer-based tools before using *Information Manager*.

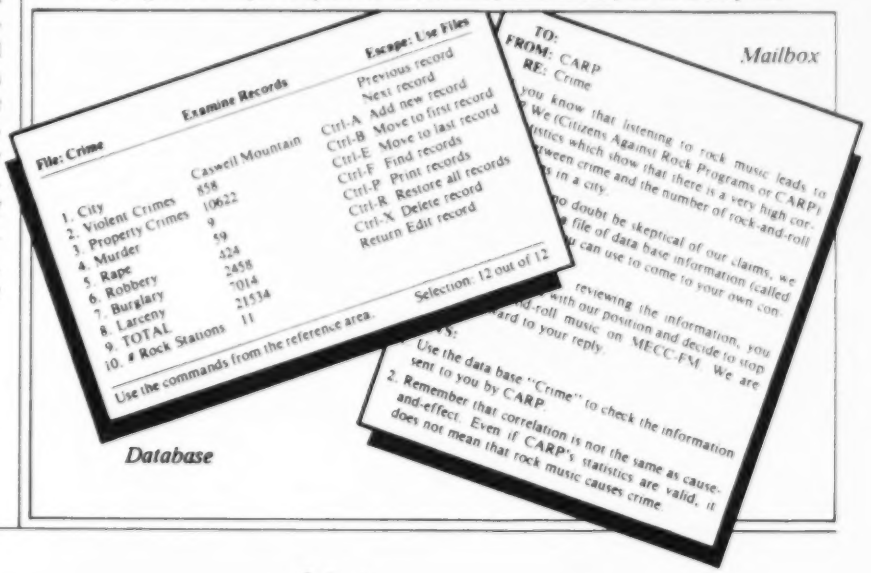
The program's major objectives are for students to analyze a problem, decide how to solve it, and use appropriate tools to find the answer. The teacher plays an active role in the simulation, teaching vocabulary and concepts and guiding discussion.

Students work individually or in pairs, each needing both a data disk (created with a program utility) and a copy of the *Information Manager* disk. Thanks to Alaska's licensing agreement with MECC, I was able to make copies of the program disk for an entire class of junior high students in the computer lab. lab.

The Tools

Students start with "Mailbox," a simulated electronic mail system through which they receive memos from the boss, the state employment commission and others with whom they, as station manager, must conduct business. Each of the seven memos presents a problem to be solved—such as balancing the station's budget—and provides hints as to which tools to use. Upon completion of the task, the students use a word processor, "Memowriter," to prepare their responses (e.g., which promotional events were the most effective in increasing the station's ratings).

"Database" contains five files. In the course of their work as station manager, students search the files for specific information, add records to an existing file, and create a new file. An outstanding feature of this data base is that all commands appear either alongside the data, in a menu, or in a help file.



© Copyright 1986 ICCE

"Calculator" is a limited-feature tool (no arithmetic functions are provided) required for the execution of two of the station manager's tasks, which finds averages and the coefficient of correlation. As in the data base, the keystrokes required for using it appear on the screen.

Ease of Use

Information Manager looks very much like *AppleWorks*. Students use the arrow keys or numbers to highlight the desired menu option and press Return to select it. The file folder borders around the screen show your progress through layers of menus: the ESC key takes you back through the layers. Commands for using each tool appear on the screen or are readily available in a help file.

Students can print out all memos and data. A "clipboard" is used to transfer data between the tools in *Information Manager* and to another MECC program, *MECC Graph*.

The excellent manual describes each component of *Information Manager* and explains in detail how to use the program. It contains step-by-step lesson plans and student handouts.

Suggested Uses

Information Manager requires students to use skills from several curricular areas. Most of the activities require students to search through one of the data bases for specific information—a research skill that might fit into both the language arts and social studies curricula. A language arts teacher could capitalize on the written communications skills used throughout the simulation, honing students' letter-writing skills and teaching students to analyze their memos with two other MECC products in their Computing and Information Collection, *MECC Editor* and *MECC Speller*.

The simulation brings us issues related to working in the communications age—such as confidentiality of electronic mail—which would make *Information Manager* appropriate for computer literacy and social studies classrooms.

Math is brought into play when students must find the coefficient of correlation between crime rate and rock music listening to appease an irate citizens' group. In another memo, the state employment commission requests figures for the mean, median and range of employees' salaries.

Problem solving or independent learning skills are used constantly. Throughout the simulation, students must carefully read each memo and plan a course of action for finding the requested information. Many of my students wanted me to tell them what to do instead of relying on the information given them by the program. As they progressed through the activities, though, they gained confidence and skill in solving problems without my assistance.

Final Notes

I spent lots of time reading the memos that students wrote to the individuals who had posed problems or requested information. I read them both to see if students had solved the problem to my satisfaction and because students wanted feedback.

The program doesn't provide students with feedback about their work. Students read a memo, find the requested information, and dispatch a reply. "How will I know if the state employment commission got my memo?" they asked me. "Was the citizens' group satisfied with what I wrote?" To provide students with feedback and to make the electronic mail system more real, I read the memos and wrote replies, which I stored on the students' data disks either in "Memowriter" or "Mailbox."

It was very time consuming to respond to all the memos, but for many students, this was their favorite part of the simulation—receiving mail from the people they'd done work for. I had fun writing in the style of the person who'd sent the original memo, using current jargon as the D.J. and a formal style as the station owner. Of course they knew that I had written the memos, but they nevertheless enjoyed the written exchange.

Do use *Information Manager* with students who've had some experience with applications software. Some of the junior high students in my group never had and were so busy struggling to gain a conceptual understanding of what was going on that they didn't participate fully in the problem-solving process.

I recommend that *Information Manager* be used in a computer lab because of the high degree of teacher involvement required to meet the objectives. I would hesitate to use *Information Manager* if only one machine or disk were available.

Information Manager's excellent screen prompts, help menus and teacher manual make it easy to use. But the instructor must be prepared to devote considerable time both in preparation outside the classroom and in doing direct teaching throughout the course of the simulation. For me, the rewards justified the effort: Students learned when and how to use several productivity tools, to pay closer attention to the instructions in the mail messages, and to use the cues on the screen instead of hollering to me for help all of the time. The simulation required lots of hard work on the part of the students, but their interest remained high throughout the course of the project.

Publisher's Reply

As Ms. Sopp states in her review, *MECC's Information Manager* is best used by students who have had prior experience with computer based tools. As a matter of fact, *MECC's Information Manager* was designed with such a prerequisite in mind. This package is from MECC's Computing and Information Collection, a group of products designed to provide students with hands-on experiences using electronic technology to process information.

Overall we found Ms. Sopp's review to be fair and accurate. We want to point out, however, that part of the power of *MECC's Information Manager* lies in its "Management Options," a feature not fully explained in the review. This feature enables teachers to create the "memos and hints" that students access through the "Mailbox." The hints may be displayed for some students or hidden from view at the teacher's discretion. Teachers may use the product's data base to create the files students use to solve the problems posed in the memos. Thus, teachers who have exhausted the "canned" problems or who find them inappropriate for their subject area may create additional scenarios for students to use.

Finally, quite apart from its use to solve problems posed by the teacher, *MECC's Information Manager* is a powerful productivity tool. The word processor and data base alone make it a useful acquisition for any computer lab.

© Copyright 1986 ICCE

WRITER'S HELPER

Conduit, The University of Iowa, Oakdale Campus, Iowa City, IA 52242

Grade/Target Age Group: Grade 8 through college

Subject: Writing

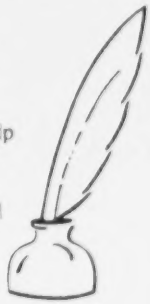
Cost: $120 for one original and one backup copy of the Program Disk, plus one unprotected Work Disk; $240 for an Educator's pack containing four additional copies of the Program Disk.

Hardware Required: Apple IIC or Apple IIe, with 80-column card; 64K RAM; single disk drive; monitor. Printer recommended.

Reviewed by: Larry Lewin, Language Arts Teacher on Special Assignment, School District 4-J, Eugene, Oregon

Writer's Helper is not a word processor: It is a series of 22 mini-programs that help student writers *plan* what to write and begin to *revise* what they have written. As any composition teacher knows, these are two major obstacles for young authors. *Writer's Helper* begins with recognizing student problems and then proceeds to help overcome them.

Writer's Helper will work well for secondary and college teachers who apply the process approach, a four-step sequence of instruction that moves students from prewriting to drafting, editing, and finally publishing. While this method is widely popular at all grade levels, teachers who use it know the demands it makes on teaching time. Because each of the mini-programs in *Writer's Helper* can be learned in a few sittings, students can use it for individual instruction during prewriting and drafting, the two stages most neglected by current software.



Prewriting Activities

The first prewriting section includes mini-programs designed to help students consider alternative topics, purposes, audiences and formats. Students can select any or all of these prewriting activities in any order.

One activity in this section is "Brainstorms," which allows students to free-write three to 10 lines without the ability to delete. After assessing original typing speed, the program marks pauses in the flow with "....." until the writer continues. I liked how this encouraged my thoughts to continue flowing. After three to 10 lines have been written, the student can save them to his/her Work Disk, print them, or transfer them to a word processor for further work.

Another activity, "Lists," allows the student to type up to 10 possible writing topics in a few words or short phrases. "Lists" is included to prevent premature closure on the prewriting stage: Listing 10 potential topics increases the chance of discovering one that will be worth pursuing. This activity could be accomplished with pencil and paper, but most students prefer using computers.

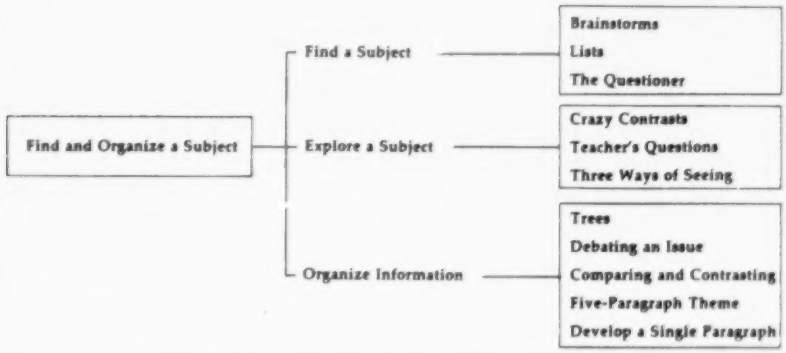
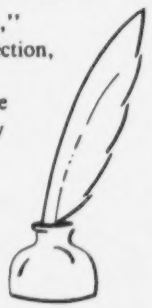


Chart of Programs in Find and Organize a Subject.

© Copyright 1986 ICCE

A third prewriting mini-program in this section, "The Questioner," lists 20 questions, one of which may help the young writer find an interesting topic, such as, "Why do some people believe in astrology?" or "What is it parents should know more about?" These may trigger excitement for some students, but I found them to be uninspiring.

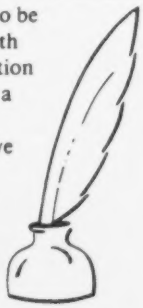
All three of these writing exercises had been used successfully for years before computers were available in the classroom. But *Writer's Helper* improves on them by merging them with a word processor. The *Bank Street Writer* or any other word processor that uses standard DOS 3.3 Applesoft text files can be used with *Writer's Helper* to expand work from these prewriting activities. This is a major strength of *Writer's Helper*.



In "Explore a Subject," another pre- writing section, "Crazy Contrasts" compares a topic choice to something completely different in order to spur creative thought. For example, after I chose "Portugal" as my topic from an earlier "List," this mini-program asked how Portugal was like "an old shoe," "an iceberg," and "Donald Duck." I didn't get too far with these comparisons, but I need to work on my creativity.

A second way a student can explore a subject is by answering up to 20 questions designed by the teacher to stress key points about a subject. Teachers will appreciate the opportunity to tailor the software to their own writing assignments.

Third, exploration of a subject can be accomplished by selecting "Three Ways of Seeing." The student types in the name of a subject, selects the appropriate label of "person, place, thing, event, idea or activity," and answers 12 to 15 questions that look at the subject: 1) in isolation; 2) as a process of change; and 3) in relation to other subjects. I was stretched in my thinking about my topic as I answered the questions. While some answers were uninteresting to me, many served to make me look at my topic in new ways, which is a key goal of prewriting.



Writer's Helper can also be used to assist students with organizing their information after they have selected a topic and collected information about it. Five mini-programs which incorporate successful writing techniques are included:

- "Trees," an outlining procedure that helps the student categorize pieces of information about a subject;
- "Debating an Issue," which asks for both pro and con sentences about a subject;
- "Comparing and Contrasting," which asks for similarities and differences between the selected topic and a related topic;
- "Five Paragraph Theme," which asks questions about a subject and reformats the sentence-answers into a rough draft; and
- "Single Paragraph," which guides the student in developing either a descriptive or argumentative paragraph.

All these instructional methods have been used in writing classes, but *Writer's Helper* enhances them by adding the power of computer memory. For example, grouping information on a topic is accomplished neatly on the screen after the student enters the group names and lists the bits of information by number beneath them. This could be done on a worksheet, of course, but the computer does it much more quickly and neatly. Likewise, structuring information about a topic into a standard five-paragraph theme can be done by students, but how nice to have the computer do it for them.

Rewriting

Writing teachers agree that computers are most effective as tools for rewriting. They greatly aid revision by quickly and neatly inserting, deleting and moving text. Of course, word processors are dumb: They can't tell the author when and why changes should be made. But *Writer's Helper* provides students with initial reactions to their writing so they can begin to consider what inserting, deleting or moving should be done. It cannot tell students whether their writing is good or bad—that job is left for the much smarter teacher—but it can help them think about their writing in an organized way and help them find errors before the teacher does.

After students have written their first drafts on a word processor, *Writer's Helper* offers 11 mini-programs to help evaluate their writing (see figure 1). The first menu choice in this section, "Paragraph and Sentence Analysis," will:

- List the first sentence of each paragraph for examination of transition words;
- Graph the number of words per paragraph for length variety consideration;
- List each sentence to show it in isolation from its neighbors for easier error detection;
- Determine readability level using the Fogg Index to compare it to the author's designated audience level; and
- Graph the number of words per sentence for length comparison (see figure 2).

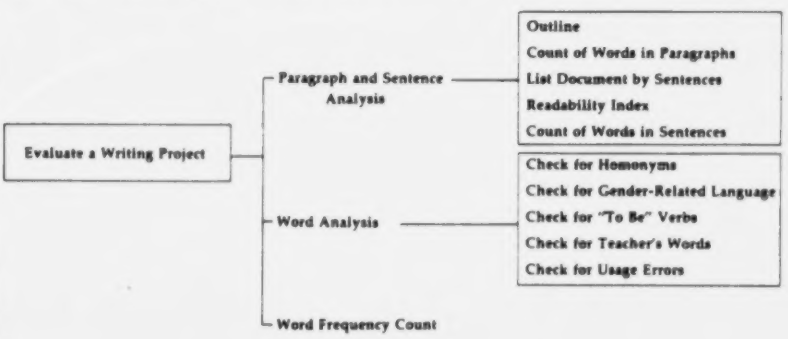


Figure 1.

© Copyright 1986 ICCE

```
          Sentence Graph
Sentence 1:    *********** 12
Sentence 2:    ***************************** 28
Sentence 3:    ********** 11
Sentence 4:    ************ 14
Sentence 5:    ********************** 22
    :
    :
Sentence 47:   ********************* 21
Sentence 48:   ************************ 23
```

Figure 2.

This section is especially useful because it quickly and effortlessly provides the writer with data that indicates a possible need for more sentence variety, removal of run-on sentences, or elimination of sentence fragments. This reminds me of the "Sentence Opening Sheet" used in the *Stack the Deck* writing textbook series; the difference is that the computer does the counting for the students, which makes it more likely they will get the data they need to improve their writing. I am impressed.

The second choice from the "Evaluate a Writing Project" menu is "Word Analysis." Five mini-program word checkers are included: 33 homonyms, 25 gender-biased words, 59 usage confusions (e.g., farther/further, accept/except), and "to be" verbs (to check for passive construction). Though the teacher cannot add words to these four checkers, all the important ones are icluded. The fifth checker does allow teacher input: words the teacher wants to reduce (over-worked, trite words) or increase (specific vocabulary). *Writer's Helper* highlights a suspect word from any of the five checkers, lists the possible corrections, and marks it, if instructed, for later search and replacement in a word processor. The program does not include a spelling checker however.

For paragraph analysis, the addition of an extra mini-program allowing the teacher input would improve *Writer's Helper*. Just as the teacher can create a list of topics or key vocabulary words for the student to consider in other mini-programs, the ability to do the same with paragraph characteristics would be helpful.

The last menu choice is "Word Frequency Count," which lists all the words in a piece of writing alphabetically, along with their number of occurrences. Because some words are inevitably popular (the, and, a), the student may exclude up to 20 words from the count, speeding up the computer's counting time. This word counter is a helpful self-evaluator: redundant words are exposed for the student before the teacher has to mark them in red ink; weak or boring words surface out of the camouflage of the sentence context; and even spelling or typographical errors are readily seen. This mini-program helps students edit their own papers independently.

Recommendation

Now that computers are increasingly available in the schools, teachers have been looking for ways to use them to assist with instruction. *Writer's Helper* is a welcome addition to writing instruction because it puts the new technology to good use: It helps students get started by guiding them through many possible subject choices in ways the teacher would, and later helps them improve by providing feedback for possible revision in ways the teacher cannot.

I am impressed with this program's awareness of the process approach to writing: It offers 11 prewriting and drafting mini-programs that use the computer effectively instead of simply offering electronic worksheets. It also has 11 mini-programs that help teach how to edit for improvement. Because the program disk is resident, having four additional disks with the Educator Pack would be useful in a lab setting to increase its independent use.

The documentation presents in a clear, readable format the reasons for each section (with source material quoted), simple student procedures, and extended activities for the teacher to consider. The author also included ways to connect the programs to the study of literature, as well as specific suggestions

© Copyright 1986 ICCE

for using the programs with writing assignments. Finally, the author wisely included a section in the manual on how to introduce a word processor to your class. He understands the writing process and has thought it through completely while designing the *Writer's Helper*.

When computer software can make the job of teaching kids how to write any easier, I will buy it. *Writer's Helper* will help the teacher of high school and college students in the areas of composition instruction. When considering *Writer's Helper* for use with your students, check the readability of text on the screen. Good readers, even middle or junior high school students, can use some or all of the 22 mini-programs. It has a 30-day money back guarantee, so check it out.

Publisher's Reply

In May, Conduit will release a combined Apple DOS 3.3/ProDOS version of *Writer's Helper* and an IBM PC, XT and jr version. Registered users of the Apple DOS 3.3 version of *Writer's Helper* can upgrade to Apple DOS 3.3/ProDOS by sending all of their program disks to Conduit by September 1, 1986. There will be no charge for the upgrade.

© Copyright 1986 ICCE

Software REVIEWS

Linda Rathje, Editor

VALIANT TURTLE ROBOT

Valiant Designs Ltd., Park House, 140 Battersea Park Road, London SW114NB, England

U.S. Distributors: Harvard Associates, Inc., 260 Beacon Street, Somerville, MA; ph. 617/492-0660; telex 880 792.
K-12 MicroMedia Publishing Inc., P.O. Box 17, Valley Cottage, NY 10989.

Grade/Target Age Group: Preprimary through adult

Subject: Robotics, Logo

Cost: $499. Includes a Valiant Turtle robot, infrared communicator, cable, and interface software. $39.95 for interface kit for additional computer models.

Hardware Required: Apple II series running Logo software; II+, IIe require an Apple Super Serial Card, Harvard Associates Serial Card or Orange Micro Grappler Card. Also available for IBM PC/PCjr, Commodore 64 and BBC-Acorn.

Reviewed by: Jim Ellickson-Brown, K-5 music teacher, Mountain View Elementary School; and Doris Mimnaugh, fourth grade teacher, Wilson Elementary School, Corvallis, Oregon.

VALIANT TURTLE
MEGAWORKS
SPELLWORKS
HABAMERGE
APPLEWORKS MAILING PROGRAM (AMP)
THE GEOMETRIC SUPPOSERS:
- **preSUPPOSER**
- **TRIANGLES**
- **QUADRILATERALS**

Build a better turtle and the world will beat a path to your door. Such was the hope of the small group of designers and engineers who, under the name Valiant Designs, developed an inexpensive, remote-controlled, turtle-shaped robot with excellent drawing accuracy. The response in England has been great since the introduction of the Valiant Turtle in 1984. Reportedly 5,000 of these mechanical terrapins are busily at work in school districts and homes throughout the country, and a newspaper has even held a nationwide contest for school children eager to win a Turtle for their classrooms.

The Valiant Turtle arrived with less fanfare in the United States with its decentralized educational decision-making environment. Nonetheless, we (a music teacher who describes himself as an incurable Logo hacker and a classroom teacher who loves Logo for its impact on the affective development of her students) noticed a small announcement in an educational computing magazine last spring and immediately decided this was the Turtle for us. We have worked with the Valiant Turtle during the past year, primarily in a fourth-grade classroom where students have a computer and work with Logo during their class and free time. Recently this class has shared its expertise and enthusiasm for the Turtle with the rest of the school, and we are as excited as they are about the impact the Valiant Turtle has made.

Description

The complete Turtle system comes in three boxes (see figure 1):

- The Valiant Turtle, which comes with 10 rechargeable batteries, two manuals, a cartoon book, a Berol pen, and a screwdriver for adjusting the wheels;
- An infra-red communicator and power adaptor; and
- An interface kit consisting of software (disk or cassette), a leaflet describing setup of the Valiant Turtle with various interface cards and dia-

© Copyright 1986 ICCE

Figure 1.

lects of Logo, and a cable to connect the communicator to the computer or interface card.

Distributor Support

The performance of the Valiant Turtle is generally excellent, and we have been happy with the support we have received from Harvard Associates. Presently they import approximately 25 Turtles at a time and run them through diagnostic tests before shipping them out. Items which fail within 90 days are replaced free of charge, and Harvard promises to provide prompt repair service at reasonable cost for Turtles which develop problems at a later date. They are happy to provide telephone assistance to talk owners through problems.

Impact on the Classroom

When we decided to get a Turtle, the most obvious application we had in mind was a way of putting Logo drawings on paper the way they were originally drawn. Important as that function is, though, the real value of the Turtle is deeper; it is an object students relate to immediately with all their senses and emotions. Its attractive design is very endearing and, from a practical point of view, enables everyone to easily understand its heading. Our students were not content to leave it on paper, but wanted it to travel throughout the classroom.

When students work with the Turtle, they pay attention to and understand the feedback given by its behavior. The impact of its concreteness—how they reacted to touching, seeing and then visualizing its actions—was not apparent until we saw them work with it. Students enjoyed editing programs they had already written to run with the Turtle. One procedure—drawing a pair of glasses—provided students time to discuss the evolution of the program as the Turtle slowly went about its job. "What is it? A circle? Can we cut it out? It could be a sun. . . . Wait! There's something else. . . ." The Turtle is always providing surprises and challenging programming ideas we have come to take for granted, such as a barn procedure where the grass extended for 600 steps—a detail we had not taken into account due to our having gotten used to wraparound behavior.

What is better than one Valiant Turtle? Two Valiant Turtles! For a few weeks this year we were able to have two Turtles in the classroom and were fascinated watching both of them simultaneously respond to commands, like synchronized swimmers or ballet dancers. A contest was held to name the pair (Eric and Erica were selected) and the students enjoyed having them play follow the leader. From a practical point of view, whenever a problem arose, we found it helpful to have more than one Turtle available for backup or isolating a problem.

On the bottom of the Turtle are switches which can be set so that four Turtles can be addressed individually; at this point this has only been implemented on the BBC-Acorn computer. This would be an excellent feature to take advantage of using the new *LogoWriter* from LCSI, which features four Turtles. Imagine the four-corner problem with four robots!

Software and Documentation

The software works well. LCSI Logo allows redefinition of primitives so that Logo commands (FD, RT, PU) will direct both the floor and screen Turtles. For Terrapin Logo new Turtle commands are included in the Valiant software; to drive the Turtle you must use an alternate set of procedures prefixed with the letter T (TFD, TRT, TBK).

It would have been nice to have more detailed documentation describing the software; knowing the purpose of the various files and the procedures they contain would aid those interested in streamlining or adapting them to a particular situation. One example, calibrating the turning angle using variables, is discussed below. Although the possibility of changing the size of the Turtle steps is mentioned in passing, there is no specific discussion of how one would do this using the distance calibration vari-

© Copyright 1986 ICCE

able (CALIBDIST), originally set so that one Turtle step equals one centimeter. We experimented with this, for example setting one step to equal one inch or 10 steps to equal a tile on the floor. We also worked on screen/Turtle proportion, adjusting the value of CALIBDIST for procedures of different sizes to make best use of the available drawing paper. One can adjust these variables using the MAKE command or create new procedures such as SETSTEPS, INCHSTEPS, or TILESTEPS. One can also turn the floor Turtle off and on (useful for debugging a procedure) as well as the screen Turtle (sometimes the screen Turtle can be distracting) with Logo commands.

The documentation provided with the interface kit gives explicit but rather lengthy directions for setting up the Valiant Turtle with each hardware and software configuration. We found this cumbersome for our students and developed a self-starting Logo file we called Turtle. Using this file, our students could work independently using a three-step procedure: 1) Turn on the computer and boot Logo; 2) plug in the communicator and turn on the Turtle; 3) type READ "Turtle.

The manual titled PENUP, provided as part of the documentation, discusses Logo in general. It also provides a sequence of activities in estimating distances and angles using the Turtle and included are some ideas for subject integration and home use.

Learning to Live with Your Turtle

We have enjoyed working with our Valiant Turtle. We did, however, encounter several problems at the beginning. Though frustrating at the time, looking back we feel that we learned from these experiences. The documentation did not address all of our questions; experimentation did.

Using our original Logo software, we found the Valiant Turtle's turning accuracy to be off considerably. Several problems presented themselves in attempting to adjust this. First, in the manual's discussion of performing this adjustment with the screwdriver, a misprint (on page 6, paragraphs 6 and 7, the word "more" should be "less" and "less" should be "more") leads one to do the opposite of what is needed. Moreover, the screwdriver provided is a little too large for the job and slightly gouges the Turtle.

We found it more effective to use a software fix for this problem. We changed the value of the variables "CALIBTURNR and "CALIBTURNL to calibrate the turning angle to our satisfaction. This is not discussed in the documentation.

The Turtle developed a problem (after it was past warranty) with a loose wheel fitting, which would inhibit the Turtle's movement. Although the manual warns that removing the plastic top of the Turtle voids the warranty, we were past that point and Harvard Associates had not yet instituted their in-house repair facility. We took the Turtle to a local toy shop where we were advised to tighten a screw which had loosened slightly and a little glue was applied to the wheel fitting. Since then the Turtle has performed flawlessly. For us, taking the top off the Turtle helped demystify its workings.

The pen provided with the Turtle is an international model Berol pen not readily available in the United States. Since no other pen seems to fit the Turtle and pens dry out quickly when their tops are left off, care should be used. Replacement pens are available through the distributors at a cost of three for $5.00. We recommend purchasing half a dozen additional pens.

Although this procedure is not mentioned explicitly in the user's manual, we found it most useful to charge the Turtle up each night after school rather than wait for it to fail in the middle of an elaborate procedure. The Turtle may be turned off and on during the day to save battery wear without interfering with its workings.

Care should be used in placement of the communicator so that the signal to the Turtle is not interrupted. One good arrangement is to have the communicator sticking out over the edge of a table; another is to have it point toward the ceiling. The rectangular shape and placement of the connecting cords on the communicator make it a bit awkward to set up; perhaps a small holder could be designed.

Conclusion

There are a variety of Turtles on the market for educators to choose from, providing among other things sound, light, collision sensors, or speech or music capabilities. While lacking these, we think the Valiant Turtle provides excellent features for teaching Logo, high accuracy, adjustability, attractive design, and remote control capability.

We have found the Valiant Turtle's presence has increased the interest level of students and teachers in Logo, facilitated the understanding of Logo commands, and changed personal interaction during Logo work, providing students more time to share and discuss their observations.

The Valiant Turtle provides many avenues for subject integration and imaginative work/play for creative students and teachers, making it ideal for those interested in closely following the Logo philosophy of experimentation and learning based on self-knowledge. Although we have had to cope with some challenges and frustrations in maintaining and fine-tuning the performance of our Valiant Turtle, we have been pleased with the product support provided by Harvard Associates. Highly recommended.

Publisher's Reply

Thank you for your review of the Valiant Turtle. Mr. Ellickson-Brown was one of the first people in the United States to get a Valiant Turtle. Since he received his a year ago, they have made their way to hundreds of classrooms across the country.

Harvard Associates has upgraded the Valiant Turtle software and documentation in response to suggestions from customers like Mr. Ellickson-Brown, which has eliminated many of the minor difficulties he experienced. We have also expanded our repair facilities and our line of Valiant Turtle accessories, which includes the Valiant Turtle tote ($59.95) to store and transport your Valiant Turtle, the Harvard Associates serial card ($75) for interfacing with Apple computers, and inexpensive replacement batteries.

© Copyright 1986 ICCE

THE GEOMETRIC SUPPOSERS:
- preSUPPOSER
- TRIANGLES
- QUADRILATERALS

Sunburst Communications, 39 Washington Avenue, Pleasantville, NY 10570-9971; ph. 800/431-1934

Grade/Target Age Group: Grades 6 through 12

Subject: Geometry

Cost: $99 (each program) with backup; $297 lab package of 10 disks.

Hardware Required: Apple IIc, IIe, II+ (64K RAM minimum); one drive.

Reviewed by: Judi Mathis and Seymour Hanfling, Graduate Students on the CI³ Project (NSF Grant #TEI 8550558), University of Oregon, Eugene, Oregon.

The Geometric Supposers provide workrooms stocked with tools specifically designed for constructing and measuring geometric shapes. The number and variety of explorations provided by these workrooms are endless. If you are an educator who teaches geometric concepts and vocabulary, or have been waiting for software to fulfill the promise of such buzzwords as "electronic sandbox" and "microworlds," then experience the Supposers.

The Geometric Supposers (there are three at this time: *The Geometric preSupposer* for lines and points, *The Geometric Supposer: Quadrilaterals* and *The Geometric Supposer: Triangles*) are tools for exploring various aspects of geometry. Students can learn specific geometric concepts (the geometric mean, area formulas, angle summations of simple polygons, etc.) and discover *new* geometric theorems. This is a powerful claim, but true. One of the students who pilot tested this software discovered a new conjecture which states how to divide any triangle into five pieces, each having equal area. (The solution is given in an article enclosed in the *Triangles* manual.)

The first time I gave students a problem to solve with the Supposers, they solved it and asked permission to pose their own problems. At that moment I realized three things:

- The program was very easy to use. These students did not need any "teaching of the software"; no valuable class time was wasted. No formal demonstration was necessary for students to begin solving problems;
- The students' basic geometry vocabulary enabled them to easily select the appropriate words and actions from the menu; and
- They *wanted* to pose their own problems. When was the last time you assigned a problem and students asked if they could make up some of their own?

Points and Lines, Triangles, and Quadrilaterals

Each program has a slightly different focus. The *preSupposer* presents the basic relationships between lines, points, polygons and circles. When the software is loaded there are no flashy titles or menus, but rather a drafting area with a short menu at the bottom (see figure 1). Using simple keystrokes students can draw lines, line segments, circles, triangles, quadrilaterals and other polygons. The Grid, unique to the *preSupposer*, is a second drafting area which provides a frame of reference for both constructions and estimation of measurement. The *preSupposer* has a Moveable Point which can be positioned anywhere in the drafting area. However, if it lands on a figure, axis or grid mark and then is moved away, it will take that pixel with it.

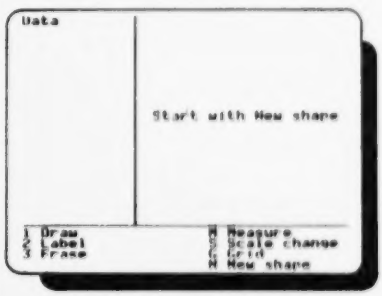


Figure 1.

The Geometric Supposer: Quadrilaterals covers (you guessed it) quadrilat-

© Copyright 1986 ICCE

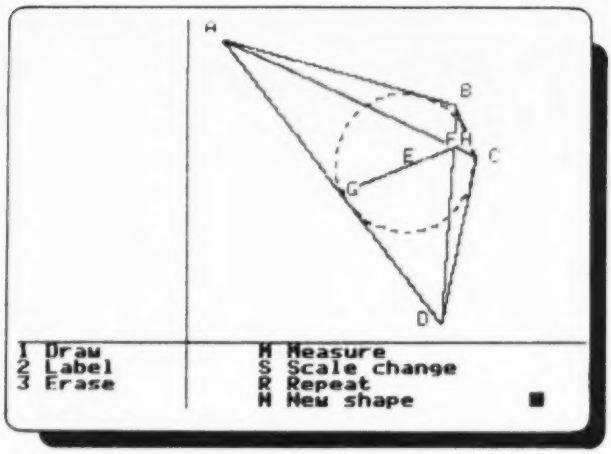


Figure 2.

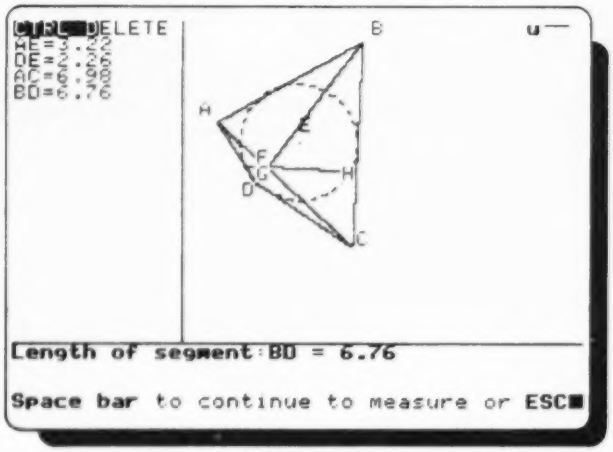


Figure 3.

erals of all types: convex, concave, inscribed in a circle, circumscribed around a circle, parallelogram, trapezoid, kite, and Your Own. There are more measurement features here than in the *pre-Supposer.* Once a multi-step construction is completed, the construction steps (up to 20) can be replicated on a second quadrilateral with only a few keystrokes. This simplifies the application of a theorem on many distinctly different examples in a short amount of time (see figures 2 and 3).

The Geometric Supposer: Triangles offers right, acute, obtuse, isosceles, equilateral and Your Own triangles. Sides or angles can be divided, labeled, measured, reflected or erased (from the screen, not from memory—they can still be accessed). Some constructions on the screen *(Geometric Supposers: Triangles* and *Quadrilaterals)* are drawn complete with arcs and construction lines just as when using a compass and straight edge (see figure 4). Then all extra lines are erased. In this way manual constructions using straight edge, compass, paper and pencil are reinforced rather than replaced.

Classroom Observations

Students worked nicely in groups of two or three using this software either in the classroom or lab. Individuals can easily use the program, but with groups the discussions are invaluable. One geometry teacher observed his students using the computer to work on some problems with the geometric mean, the focus of the test to be given the next school day. Although all the students had read the textbook definition, they were concentrating on a working definition—an intuitive notion of the concept. He later added a question to the next day's test: "Give your definition of the geometric mean."

Each program can be used as a demonstration tool for introducing a concept, applying a theorem to a number of different shapes, reinforcing a concept, posing interesting explorations, clarifying a construction, or making a lesson more visual. Now I don't have to take 10 or 15 extra minutes before class to draw a complex construction for a two-minute discussion. Now I can demonstrate the application of every theorem about triangles using many different examples: isosceles, obtuse, right and equilateral. For example, after drawing the median of a triangle I asked students to determine how the two smaller triangles compared, what could be deduced about those two smaller triangles, and if this always worked. With the Geometric Supposers, student exploration does not end with teacher demonstration.

For the teacher, exploring alternative ways of teaching geometry can begin with tools like these. The software encourages, almost entices, teachers and students to pose and attempt to prove geometric hypotheses. Suddenly everyone seems to ask, "What do you suppose happens when . . . ?" or "What if . . . ?" This software supports the premise that when students have control over the examples or applications of a concept they will have a deeper intuitive understanding of it than when a teacher

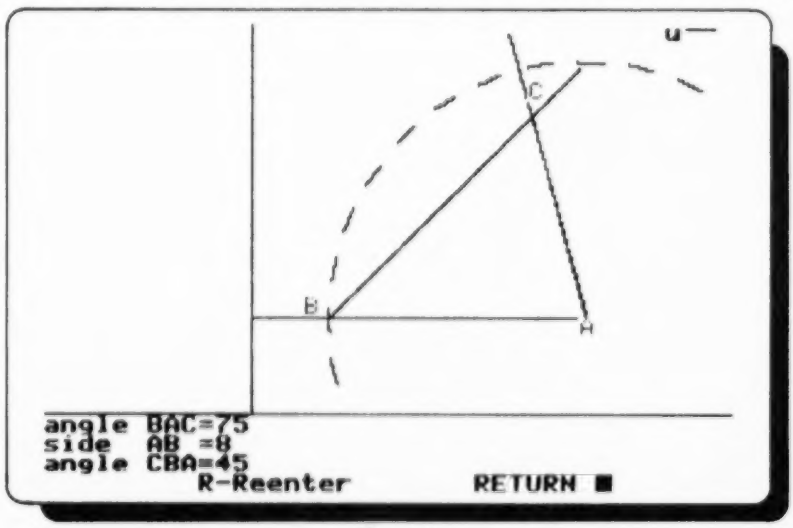


Figure 4.

© Copyright 1986 ICCE

presents an idea and supports it with a couple of examples. Two young ladies asked, "Are there any more fun questions like the one about trisecting an angle?" As a teacher I was trained to motivate students, not to handle enthusiasm for impossible problems. The manual includes many stimulating problems for further consideration.

Strengths

The three Supposers provide an environment which encourages geometric curiosity among computer novices as well as experts. Program features creating this environment include:

- Simple screens and menus allow students to quickly begin exploring;
- A variety of menu options allow a vast range of geometric explorations;
- Automatic replication of the same constructions on quadrilaterals and triangles is possible;
- Editing features are good;
- Measurements can be made of lengths and areas (within a certain error due to the discrete organization of the computer);
- Uncluttered constructions make demonstrations clearer;
- Manual has clear explanations for all menus and options;
- Manual provides excellent ideas for teachers;
- Reference Cards are thorough and useful;
- Program fully loads into a 64K machines—no disk accessing; and
- Sunburst allows multiple loadings.

Limitations

Constructions are not always easy when designing Your Own triangle or quadrilateral. If the first angle chosen is too great or the length too long, no shape results. Even if the object or label appears to fit, if it is too close to a boundary the program does not acknowledge it. Using Random Point is not the same as being able to place any point you need. Scale change does not have enough range to suit all the users' possible demands.

The use of a mouse, joystick or drafting pad could eliminate some of these problems and open up new possibilities. A printing option would aid both students and teachers.

Recommendation

Sunburst's Geometric Supposers are exciting and significant additions to the world of mathematics software for both students and teachers. The Supposers provide an exploratory environment where students can experience and develop an intuitive understanding of geometric concepts.

These programs complement a variety of teaching styles—from lecture format to the discovery approach. With a large-screen monitor a teacher can quickly construct and manipulate the figures needed for a given lecture. Students are motivated to explore and thus learn geometric concepts. The software encourages the higher level thinking skills involved in formulating and testing hypotheses. We recommend teachers seriously consider purchasing this set. If you can only afford one at this time, purchase the *preSupposer*. It provides many of the functions of the other two tools.

One young man's parting words: "Is it legal to use that piece of software at home?" Yes! How might the future of teaching geometry change if students had access to these programs on a regular basis?

The reviewers would like to thank the math teachers of Churchill High School in Eugene, OR. Special thanks go to geometry teacher Don Cataldo and his inquisitive students.

Publisher's Reply

Sunburst wants to thank *TCT* for its thoughtful review and for the recognition of the program design of Judah L. Schwartz and Michael Yerushalmy through the Center for Learning Technology at the Education Development Center. For the readers' information, a fourth program, *Geometric Supposers: Circles*, will be available in the fall, along with a student workbook to accompany the program *Geometric Supposers: Triangles*. Sunburst also provides a Geometric Supposer videotape that is ideal for classroom presentations and demonstrations. The videotape is available on a 30-day loan.

© Copyright 1986 ICCE